



TUGAS AKHIR - TJ141502

**NAVIGASI UNTUK OLAHRAGA BERSEPEDA DAN
BERLARI BERDASARKAN KUALITAS UDARA**

Faishol Lukman Hakim
NRP 2913 100 042

Dosen Pembimbing
Arief Kurniawan, ST., MT.
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TJ141502

CYCLING AND RUNNING EXERCISE NAVIGATION BASED ON AIR QUALITY

Faishol Lukman Hakim
NRP 2913 100 042

Advisor
Arief Kurniawan, ST., MT.
Dr. Eko Mulyanto Yuniarno, ST., MT.

Departement of Computer Engineering
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017

LEMBAR PENGESAHAN

Navigasi untuk Olahraga Bersepeda dan Berlari Berdasarkan Kualitas Udara

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Faishol Lukman Hakim (NRP: 2913100042)

Tanggal Ujian : 12 Juli 2017

Periode Wisuda : September 2017

Disetujui oleh:

Arief Kurniawan, ST., MT.
NIP: 197409072002121001

(Pembimbing I)

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP: 196806011995121009

(Pembimbing-II)

Dr. Surya Sumpeno, ST., M.Sc.
NIP: 196912091997031002

(Penguji I)

Eko Pramunanto, ST., MT.
NIP: 196612031994121001

(Penguji II)

Ahmad Zaini, ST., MT.
NIP: 197504192002121003

(Penguji III)

Mengetahui
Kepala Departemen Teknik Komputer

Dr. I Ketut Eddy Purnama, S.T., M.T.
NIP. 196907301995121001

DEPARTEMEN
TEKNIK KOMPUTER

ABSTRAK

Nama Mahasiswa : Faishol Lukman Hakim
Judul Tugas Akhir : Navigasi untuk Olahraga Bersepeda dan
Berlari Berdasarkan Kualitas Udara
Pembimbing : 1. Arief Kurniawan, ST., MT.
2. Dr. Eko Mulyanto Yuniarno, ST., MT.

Kondisi udara pada jalanan yang telah terpapar polusi membuat jalan tersebut tidak baik untuk berolahraga. Masyarakat yang ingin berolahraga tidak mengetahui daerah jalan yang memiliki kualitas udara yang baik. Aplikasi ini bertujuan untuk mencari jalur dengan kualitas udara yang baik lalu menggambarkan dan mewarnainya berdasarkan kategori ISPU (Indeks Standar Pencemaran Udara) serta memandu pengguna dalam melewatinya. Metode pencarian jalur menggunakan algoritma *dijkstra* terhadap data koordinat persimpangan jalan yang telah dikumpulkan sebelumnya. Sistem navigasi yang memandu pengguna menggunakan perintah suara dan papan tanda informasi arah selanjutnya. Dari hasil pengujian lapangan, didapatkan bahwa seluruh alur sistem pada aplikasi ini dapat berjalan dengan baik. Layanan lokasi paling akurat adalah *Fused Location Provider* daripada *Network* dan *GPS*. Hasil pencarian jalur dari Sistem Tugas Akhir didapatkan total indeks kualitas udara lebih baik daripada Sistem *Google Maps*. Pada Fitur penentuan arah persimpangan, metode perubahan vektor didapatkan hasil 100% akurat sesuai dengan keadaan aslinya.

Kata Kunci : Sistem Navigasi, Kualitas Udara, Pencarian Jalur

ABSTRACT

Name : Faishol Lukman Hakim
Title : *Cycling and Running Exercise Navigation
Based on Air Quality*
Advisors : 1. Arief Kurniawan, ST., MT.
2. Dr. Eko Mulyanto Yuniarno, ST., MT.

Air conditions on roads that have been contaminated by pollution make the road is not good for exercise. Civillian who want to do exercise have no idea about the road areas that have good air quality. This application aims to find a path with good air quality and then draw and color it based on the category of standard air pollution index. This application also guides uses through the path. Pathfinding using Dijkstra's algorithm to the collected junction coordinate data. The navigation system that guides the user using voice commands and signposts for further directions. From the result of application testing inform that the whole system goes well. Pathfinding result of the system inform that air total of air quality index are better than the result of Google Maps System. Feature which is determine the direction of the junction have 100% accuracy using the result of vector change method.

Keywords : Navigation System, Air Quality, Pathfinding

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Navigasi Untuk Bersepeda dan Berlari Berdasarkan Kualitas Udara**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, Fakultas Teknologi Elektro ITS, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ibu, Ayah, Almarhum Farid Naufal Hadziq dan keluarga yang telah memberikan dorongan mental, spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Kepala Departemen Teknik Komputer ITS Dr. I Ketut Eddy Purnama, ST., MT.
3. Bapak Arief Kurniawan, ST., MT. dan Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. atas bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer ITS, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman *B201-crew* , teman-teman mahasiswa Teknik Komputer ITS, serta teman-teman alumni MBI Amani Ummah generasi 5.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juni 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak	iii
Abstract	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR KODE	xvii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	3
1.5 Sistematika Penulisan	3
2 TINJAUAN PUSTAKA	5
2.1 Olahraga Bersepeda dan Berlari	5
2.2 Navigasi	5
2.2.1 Sistem Navigasi	5
2.2.2 Sistem Navigasi pada Tugas Akhir	6
2.3 <i>Graph</i>	6
2.4 Ppm	7
2.5 Algoritma <i>Dijkstra</i>	7
2.6 <i>GPS</i>	8
2.6.1 Cara Kerja <i>GPS</i>	9
2.6.2 Pengaplikasian <i>GPS</i>	10
2.6.3 <i>Latitude</i> dan <i>Longitude</i>	10
2.6.4 Penentuan Posisi pada Android	12
2.7 Formula <i>Haversine</i>	13
2.8 <i>Google API</i>	14

2.8.1	<i>Google Android API</i>	14
2.8.2	<i>Google Maps API</i>	15
2.9	Indeks Standar Pencemaran Udara	16
2.10	Akibat dari Kadar CO yang Berlebihan pada Tubuh	18
3	DESAIN DAN IMPLEMENTASI SISTEM	19
3.1	Desain Sistem	19
3.1.1	<i>Data Flow Diagram</i>	19
3.1.2	<i>Entity Relationship Diagram</i>	20
3.2	Alur Kerja Sistem	21
3.3	Pengumpulan Koordinat Persimpangan	22
3.4	Mendapatkan Lokasi Pengguna	25
3.5	Menentukan Titik Awal dan Tujuan	26
3.5.1	Perhitungan Jarak Antar Koordinat Menggunakan <i>Haversine Formula</i>	26
3.5.2	Mencari Indeks Koordinat Terdekat	27
3.6	Pencarian dan Penggambaran Jalur	28
3.6.1	Bobot <i>Graph</i>	28
3.6.2	Pencarian Jalur Menggunakan Algoritma <i>Dijkstra</i>	30
3.6.3	Penggambaran dan Pewarnaan Jalur Pada Aplikasi	31
3.6.4	Sistem Navigasi	32
3.6.5	Kondisi Saat Seluruh Jalan Berbahaya	34
3.7	Implementasi Antarmuka Aplikasi	34
3.7.1	Antarmuka Pemilihan Lokasi Awal dan Tujuan	35
3.7.2	Antarmuka Penggambaran dan Pewarnaan Jalur	36
3.7.3	Antarmuka Sistem Navigasi	37
4	PENGUJIAN DAN ANALISA	39
4.1	Metode Pengujian	39
4.2	Perbandingan Metode Layanan Lokasi	40
4.2.1	GPS Provider	40
4.2.2	<i>Network Provider</i>	40
4.2.3	<i>Fused Location Provider</i>	42
4.3	Pencarian Jalur	44
4.3.1	Total Jarak yang Didapatkan	44

4.3.2	Total Indeks Kualitas Udara yang Didapatkan	45
4.4	Penentuan Arah Pada Persimpangan	46
4.5	Waktu Tempuh Setiap Alur Aplikasi	46
4.5.1	Penentuan Tujuan	47
4.5.2	Pencarian Jalur	47
4.6	Pengujian Aplikasi Lewat Kuisioner	48
5	PENUTUP	51
5.1	Kesimpulan	51
5.2	Saran	52
	DAFTAR PUSTAKA	53
	LAMPIRAN	55
	Biografi Penulis	55

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1	<i>Graph</i> dengan 6 <i>node</i> dan 6 <i>edge</i> [1]	6
2.2	Segmen pada <i>GPS</i> [2]	9
2.3	<i>Latitude</i> dan <i>Longitude</i> [3]	11
2.4	Visualisasi hukum <i>Haversine</i>	14
2.5	<i>URL</i> untuk akses <i>Google Maps Directions API</i>	15
3.1	Data Flow Diagram	20
3.2	<i>Entity Relationship Diagram</i>	20
3.3	Alur Kerja Sistem	21
3.4	Gambaran Koordinat Persimpangan Jalan yang Dikumpulkan	22
3.5	Hasil Koordinat Persimpangan Jalan yang berhasil terkumpul	23
3.6	Alur Pengumpulan Koordinat Persimpangan	24
3.7	Mendapatkan lokasi pengguna	25
3.8	Alur penentuan titik tujuan	27
3.9	Gambaran Polusi Udara pada Jalan	28
3.10	Data <i>Graph</i> Jalan yang Telah Dikumpulkan Dipengaruhi oleh Jarak dan Kualitas Udara	29
3.11	Hasil Pencarian jalur menggunakan bahasa <i>PHP</i> dari data kualitas udara	31
3.12	Contoh Penggambaran dan Pewarnaan Jalur	32
3.13	Penggambaran dan Pewarnaan Jalur	32
3.14	Dialog Peringatan Tentang Kondisi Jalanan yang Buruk	34
3.15	Antarmuka Menampilkan Lokasi Pengguna dan Memilih Tujuan	35
3.16	Penggambaran dan Pewarnaan Jalur	36
3.17	Antarmuka Sistem Navigasi	37
3.18	<i>Flowchart</i> Navigasi Arah Persimpangan dan Suara	38
4.1	Perbandingan Ketiga Selisih Layanan Lokasi	43
4.2	Perbandingan Selisih <i>GPS</i> dan <i>Fused Location</i>	44
4.3	Perbandingan Total Jarak yang Didapatkan dari Dua Tipe Bobot	45

4.4	Perbandingan Total Indeks Kualitas Udara yang Didapatkan dari Sistem Tugas Akhir dan Sistem Tugas Akhir	45
-----	---	----

DAFTAR TABEL

2.1	Klasifikasi Warna Berdasarkan Indeks Kualitas Udara	17
2.2	Parameter-Parameter Dasar Untuk Indeks Standar Pencemar Udara (ISPU) Dan Periode Waktu Pengukuran	17
2.3	Data Gejala yang Didapatkan Selama Selang Waktu dan Konsentrasi Terpapar Gas CO [4]	18
3.1	Koordinat yang dikumpulkan	24
3.2	Efek dari prioritas berbeda	25
3.3	Konstanta Prioritas Indeks Kualitas Udara	29
3.4	Penentuan Arah Persimpangan	34
3.5	Klasifikasi Warna Jalur Berdasarkan Kategori ISPU	37
4.1	Perbandingan dengan Menggunakan <i>GPS Provider</i> .	41
4.2	Perbandingan dengan Menggunakan <i>Network Provider</i>	41
4.3	Perbandingan dengan Menggunakan <i>Fused Location Provider</i>	42
4.4	Pengujian Penentuan Arah Pada Persimpangan . . .	46
4.5	Hasil Pengujian Waktu Eksekusi Alur Penentuan Tujuan	47
4.6	Hasil Pengujian Waktu Eksekusi Alur Pencarian Jalur	48
4.7	Daftar Pernyataan Dalam Kuisisioner	49
4.8	Presentase Responden Terhadap Pernyataan Pada Kuisisioner	49

Halaman ini sengaja dikosongkan

DAFTAR KODE

2.1	Izin menggunakan <i>Fine Location</i>	12
2.2	Izin menggunakan <i>Coarse Location</i>	13
3.1	Mencari koordinat persimpangan jalan yang paling dekat dengan suatu koordinat menggunakan formula haversine untuk klasifikasi	26
3.2	Mencari indeks koordinat persimpangan paling dekat dengan koordinat pilihan pengguna	27
3.3	Algoritma Dijkstra untuk Pencarian Jalur	30

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Kondisi udara jalan raya hingga saat ini sangat memprihatinkan, terutama di daerah kota metropolitan dan daerah industri yang memang menyumbang polusi yang cukup banyak. Tidak seimbang nya penambahan jumlah kendaraan dengan sarana jalan yang tersedia mengakibatkan pada beberapa ruas jalan yang menjadi jalur utama kendaraan umum terjadi kemacetan. Kemacetan kendaraan bermotor ini memberikan dampak negatif berupa pencemaran udara. Penggunaan bahan bakar yang dipergunakan sebagai penggerak bagi kendaraan, sistem ventilasi mesin dan yang terutama adalah gas buangan dari knalpot hasil pembakaran bahan bakar yang merupakan pencampuran dari ratusan gas dan aerosol serta limbah gas buangan industri juga menjadi penyebab utama keluarnya berbagai pencemaran[5].

Berada di persimpangan ketika lampu lalu lintas berwarna merah bisa memperparah masalah kesehatan karena membuat orang semakin terpapar polusi dari asap kendaraan. Bahaya pada mereka yang berada di luar kendaraan atau di pinggir jalan lebih besar lagi. Organisasi Kesehatan Dunia (WHO) memperingatkan polusi udara di luar ruangan sama berbahayanya seperti asap rokok.

Pakar kesehatan *Keith Prowse* dari *British Lung Foundation* di Inggris menyarankan untuk menunggu kondisi udara diluar agak bersih atau setidaknya berolahraga didalam ruangan. Padahal berolahraga di dalam ruangan tidak menghasilkan rangsangan untuk mengeluarkan endorfin, atau menghilangkan stres, dan tidak cocok untuk jenis olahraga luar ruangan seperti berlari dan bersepeda.

Pengembangan aplikasi berbasis navigasi pada saat ini lebih fokus pada pencarian jalur terdekat, kepadatan lalu lintas, jejaring sosial, dan lokasi wisata. Sampai saat ini belum ada inovasi tentang navigasi yang menyentuh bidang kesehatan, terutama navigasi jalur

udara sehat.

1.2 Permasalahan

Berdasarkan kondisi udara di area perkotaan dan alur perkembangan aplikasi navigasi dan metode pencarian jalur, maka dapat dirumuskan permasalahan sebagai berikut:

1. Kebiasaan masyarakat perkotaan dalam berolahraga maupun beraktivitas di luar ruangan pada area yang kualitas udaranya buruk mengakibatkan kondisi kesehatan tubuh terutama paru-paru terancam dikarenakan penghirupan udara yang tidak sehat selama berolahraga dan beraktivitas di luar ruangan.
2. Kurangnya pengetahuan masyarakat tentang kondisi udara pada lingkungan sekitar yang baik untuk berolahraga.
3. Pemanfaatan sistem navigasi pada aplikasi android hingga saat ini belum ada yang memanfaatkan navigasi untuk menyelesaikan masalah bagi masyarakat yang ingin berolahraga maupun beraktivitas di luar ruangan tanpa takut berakibat buruk bagi kesehatan paru-paru mereka
4. Aplikasi navigasi yang pada umumnya hanya mempertimbangkan jarak dan tingkat kemacetan sebagai bobot, sehingga perlu ditambahkan indeks kualitas udara dari setiap *edge* pada jalan.

Dengan adanya permasalahan-permasalahan tersebut, maka aplikasi yang akan dibuat bertujuan untuk mencari jalur dengan kualitas udara terbaik dan memandu pengguna melewati jalur tersebut dengan data kualitas udara jalanan yang sudah ada . Proses pembuatan aplikasi yang dilakukan antara lain pengumpulan koordinat persimpangan daerah Surabaya, mendapatkan lokasi pengguna, menentukan titik awal dan tujuan, pencarian dan visualisasi jalur, serta sistem navigasi.

1.3 Tujuan

Tujuan utama dari tugas akhir ini adalah membuat aplikasi navigasi berdasarkan kualitas udara untuk:

1. Mengumpulkan koordinat persimpangan jalan daerah Surabaya
2. Mencari jalur dengan kualitas udara terbaik

3. Menggambarkan jalur yang didapatkan dari pencarian jalur
4. Menyajikan visualisasi kadar udara pada jalan yang didapatkan menggunakan warna berdasarkan indeks standar pencemaran udara
5. Memandu pengguna melewati jalur menggunakan perintah suara

1.4 Batasan masalah

Untuk memfokuskan permasalahan yang akan diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut diantaranya adalah:

1. Aplikasi ini dijalankan pada perangkat bergerak berbasis android.
2. *API* untuk menampilkan peta pada aplikasi dan mendapatkan data koordinat persimpangan dan jarak pada *database* menggunakan *Google Maps API*.
3. Nilai indeks kualitas udara disetiap jalan diambil dari basis data yang sudah ada.
4. Pengambilan data koordinat persimpangan dan pengujian aplikasi yang hanya dilakukan di daerah Surabaya.
5. Sistem pencarian jalur memberikan jalur hanya satu jalur teraman yang ditampilkan
6. Diasumsikan titik awal dan titik akhir berbeda.

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Tinjauan Pustaka
Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam

penelitian, yaitu informasi terkait fitur *GPS*, Pencarian jalur, *Google Maps API*, sistem navigasi dan teori-teori penunjang lainnya.

3. BAB III Perancangan Sistem dan Implementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah hingga menghasilkan visualisasi pencarian jalur dan sistem navigasi. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implentasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Keseluruhan alur aplikasi akan diuji dalam bab ini mulai dari penentuan lokasi pengguna, pencarian jalur, hingga sistem navigasi

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Olahraga Bersepeda dan Berlari

American Heart Association merekomendasikan untuk para pemula yang belum pernah berolahraga seperti bersepeda dan berlari untuk melakukannya selama 150 menit per minggu. Jika sudah mulai terbiasa disarankan melakukannya 75 menit per minggu. *American Heart Association* juga mencatat bahwa dua atau tiga latihan 10 atau 15 menit setiap hari dapat memberikan manfaat daripada latihan 30 menit tunggal.

Jika seseorang secara rutin dan teratur berolahraga pada daerah dengan polusi udara. Hal ini akan berdampak buruk terhadap kesehatannya. Terutama untuk orang yang mempunyai penyakit paru-paru kronis, jantung atau diabetes. Saat orang melakukan olahraga. Bahkan dengan intensitas yang rendah, maka ia akan bernafas sepuluh kali lipat lebih banyak dibandingkan dengan saat sedang istirahat. Saat berolahraga orang cenderung akan menarik nafas dalam ke dalam paru-paru dan bernafas melalui mulut. Sehingga melewati penyaringan saluran di hidung. Inilah yang menjadi faktor penyebab meningkatnya kontak tubuh terhadap polusi, kombinasi olahraga dengan keadaan udara yang berpolusi menimbulkan resiko.

Olahraga seperti bersepeda dan berlari yang baik tidak beracut pada seberapa jauh atau dekat olahraga tersebut dilakukan melainkan lama waktu dan udara yang dihirup.

2.2 Navigasi

Berikut adalah definisi dari sistem navigasi yang umum dengan sistem navigasi pada tugas akhir ini:

2.2.1 Sistem Navigasi

Sistem navigasi kendaraan adalah perangkat navigasi berkendaraan modern yang digunakan untuk memandu perjalanan dari suatu tempat ke suatu tujuan tertentu, dengan menggunakan per-

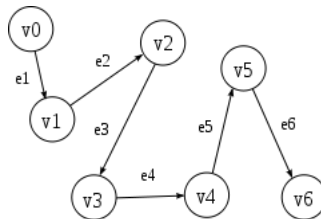
angkat peta digital dan informasi posisi dengan menggunakan satelit *GPS*.

2.2.2 Sistem Navigasi pada Tugas Akhir

Sistem navigasi pada tugas akhir ini adalah pemandu perjalanan pengguna melewati jalur yang didapatkan dari pencarian jalur yang memprioritaskan kualitas udara yang baik, meskipun menempuh jarak yang lebih jauh. Akan tetapi sistem hanya akan menampilkan satu opsi jalan yang teraman tidak ada jalan alternatif.

2.3 Graph

Graph dalam ilmu komputer merupakan tipe data yang didefinisikan sendiri sesuai kebutuhannya dalam mengolah graf hal tersebut dapat disebut sebagai *abstract data type*. Implementasinya berupa *undirected graph* dan *directed graph*. Struktur data *graph* terdiri dari *vertices* atau *nodes* dan *edge*. Struktur data *graph* juga memiliki nilai dari setiap *edge* yang dinamakan *cost*. *Directed graph* terdiri dari *node* yang berpasangan dan busur sebagai pemberi arah. *Undirected graph* terdiri dari *node* yang berpasangan dan *edge* sebagai penghubung antar *node* dengan nilai. Gambar 2.1 adalah contoh dari *graph* dengan enam *node* dan enam *edge*.



Gambar 2.1: *Graph* dengan 6 *node* dan 6 *edge* [1]

Suatu *graph* dapat dinyatakan sebagai berikut:

$$\begin{aligned}
 G &= \langle V, E \rangle \\
 V &= v_1, v_2, \dots, v_n \\
 E &= e_1, e_2, \dots, e_n
 \end{aligned}
 \tag{2.1}$$

Keterangan:

V = himpunan tidak-kosong dari simpul-simpul

E = himpunan sisi yang menghubungkan sepasang simpul

1. *Node* adalah unsur pembangun *graph*, dalam penelitian ini *node* merupakan representasi dari koordinat persimpangan jalan, berisikan *latitude* dan *longitude*.
2. *Edge* merupakan penghubung antara *node* satu dengan *node* lain. Tiap *edge* berisikan nilai yang menentukan prioritas dalam pencarian jalur.

2.4 Ppm

Ppm atau *Part per Million* adalah satuan konsentrasi yang sering dipergunakan dalam di cabang Kimia Analisa. Satuan ini sering digunakan untuk menunjukkan kandungan suatu senyawa dalam suatu media senyawa larutan atau udara[6].

Dalam tugas akhir ini konsentrasinya merupakan perbandingan antara berapa bagian senyawa CO dalam satu juta bagian suatu sistem. Sama halnya dengan prosentase yang menunjukkan bagian per seratus.

2.5 Algoritma *Dijkstra*

Algoritma *Dijkstra* merupakan salah satu varian dari algoritma *greedy*, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Algoritma *Dijkstra* membagi *node* menjadi dua kategori pada *graph*, *node Permanent* (P) dan *node Temporary* (T). Kumpulan *node* P adalah kumpulan *node* yang memiliki rute terbaik dari *node* awal hingga *node* tersebut, sisa dari *node* adalah *node* T. Inti dari proses algoritma adalah untuk mengubah kumpulan *node* T menjadi kumpulan *node* P.

Algoritma *Dijkstra* melibatkan pemasangan label pada *vertex*. Misalkan $L(v)$ menyatakan label dari *vertex* V. Pada setiap pembahasan, beberapa *vertex* mempunyai label sementara dan yang lain mempunyai label tetap. Misalkan T menyatakan himpunan *vertex* yang mempunyai label sementara. Dalam menggambarkan algoritma tersebut kumpulan *vertex* yang mempunyai label tetap akan dilingkari. Selanjutnya jika $L(v)$ adalah label tetap dari *vertex* V, maka $L(v)$ merupakan label sementara ke tetap. Pada bagian ini $L(z)$ merupakan panjang kan merupakan lintasan terpendek dari a ke z. P pada algoritma *Dijkstra* *node* digunakan, karena algoritma *Dijkstra* menggunakan diagram pohon untuk penentuan jalur

lintasan terpendek dan menggunakan *graph* yang berarah.

Rumusan Algoritma *Dijkstra* [7] :

$V(G)$	=	$\{v_1, v_2, v_3, \dots, v_n\}$
L	=	Himpunan <i>node</i> $V(G)$ yang sudah terpilih dalam jalur terpendek
$D(j)$	=	Jumlah bobot jarak terkecil dari v_i ke v_j
$W(i, j)$	=	Bobot garis dari <i>node</i> ke <i>node</i> v_j
$W^*(1, j)$	=	Jumlah bobot jarak terkecil dari v_1 ke v_j

Secara formal, algoritma Dijkstra untuk mencari jarak terpendek adalah:

1. $L = \{ \}$
2. $V = \{ v_2, v_3, \dots, v_n \}$
3. Untuk $i = 2, 3, \dots, n$, lakukan $D(i) = w(1, i)$
4. Selama $v_n \in L$ lakukan :
 - a. Pilih *node* $v_k \in V-L$ dengan $D(k)$ terkecil $L = L \cup \{v_k\}$
 - b. Untuk setiap $v_j \in V-L$ lakukan:
Jika $D(j) > D(k) + w(k, j)$ maka ganti $D(j)$ dengan $D(k) + w(k, j)$
untuk setiap $v_j \in V$, $w^*(1, j) = D(j)$.

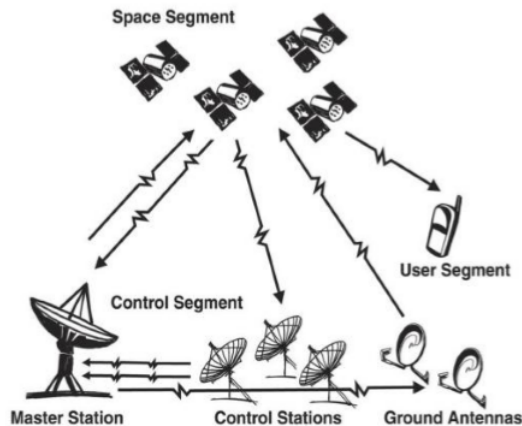
2.6 GPS

GPS atau *Global Positioning System* adalah sistem berbasis satelit yang digunakan dalam sistem navigasi untuk mendapatkan posisi dimanapun berada didalam bumi. *GPS* komersial dengan keakurasian posisi sebesar kurang lebih 15 meter dan berfungsi untuk menentukan posisi lokasi awal dan lokasi tujuan. Posisi yang diperoleh adalah posisi yang benar terhadap sistem koordinat bumi. *GPS* terdiri dari satelit-satelit, stasiun pengawas dan kontrol, serta Penerima *GPS*. Penerima *GPS* mendapatkan informasi yang dikirimkan oleh satelit pemancar dan menggunakan metode triangulasi untuk menghitung lokasi pengguna secara pasti. Fungsi *GPS* juga bervariasi , seperti:

1. untuk menentukan posisi
2. untuk navigasi dari suatu lokasi ke lokasi lain
3. untuk membuat peta digital
4. untuk menentukan jarak antara dua titik

2.6.1 Cara Kerja *GPS*

Satelit-satelit *GPS* diluar bumi terus menerus mengorbit bumi. Satelit tersebut dibekali dengan jam atomik dan sinyal transmisi radio yang berisikan posisi aktual, waktu, dan informasi lainnya. Sinyal radio yang ditransmisikan dari satelit dipantau dan dikoreksi oleh stasiun kontrol yang dikirim kembali ke satelit menggunakan antenna. Sinyal radio dari satelit diterima oleh penerima *GPS*. Penerima *GPS* hanya membutuhkan tiga satelit untuk menggambar secara kasar dalam posisi 2D, yang tidak terlalu akurat. Idealnya 4 atau lebih satelit dibutuhkan untuk mendapatkan gambar dalam posisi 3D [8].



Gambar 2.2: Segmen pada *GPS*[2]

1. Segmen Luar Angkasa : satelit-satelit yang mengorbit bumi
2. Segmen Pengontrol : stasiun pemantau dan kontrol
3. Segmen Pengguna : penerima *GPS*.

Secara garis besar penentuan posisi dengan *GPS* ini dibagi menjadi dua metode:

1. Metode Absolut atau *point positioning*, menentukan posisi dengan menggunakan lebih dari satu penerima saja. Ketelitian hanya dalam beberapa meter dan umumnya hanya diperuntukkan bagi keperluan navigasi.

2. Metode Relatif atau *differential positioning*, menentukan posisi dengan menggunakan lebih dari sebuah penerima. Satu *GPS* dipasang pada lokasi tertentu dimuka bumi dan secara terus menerus menerima sinyal dari satelit dalam jangka waktu tertentu dijadikan sebagai referensi lainnya. Metode ini menghasilkan posisi berketelitian tinggi dan diaplikasikan untuk keperluan survei geodesi ataupun pemetaan yang memerlukan ketelitian tinggi.

2.6.2 Pengaplikasian *GPS*

Karena fungsinya untuk mendapatkan data koordinat lokasi perangkat, teknologi *GPS* memiliki berbagai macam pengaplikasian pada bidang-bidang tertentu, diantaranya:

1. **Sipil**

Dalam bidang sipil teknologi GPS biasa digunakan untuk navigasi mobil, berkendara, kapal, dan pesawat. Seluruh Penerima *GPS* bagi warga sipil dapat berfungsi diatas 18 *km* (60,000 kaki) *altitude* dan 515 *m/s*(1,000 knots).

2. **Militer**

Dalam bidang militer *GPS* digunakan untuk menemukan lokasi dengan segala kondisi, meskipun di kegelapan atau bahkan medan yang tidak diketahui, dan untuk mengarahkan pasukan serta pengiriman persediaan. Dengan akselerasi 12,000 *g* atau sekitar 118 *km/s*² dan ketelitian 155-millimeter (6.1 in).

2.6.3 *Latitude* dan *Longitude*

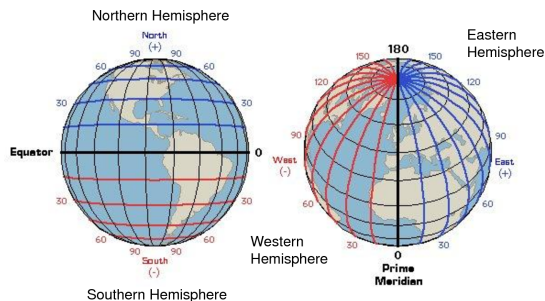
Longitude dan *Latitude* adalah suatu sistem koordinat geografis yang digunakan untuk menentukan lokasi suatu tempat di permukaan bumi. *Latitude* atau garis lintang adalah garis yang menentukan lokasi berada di sebelah utara atau selatan ekuator. Garis lintang diukur mulai dari titik 0 derajat dari khatulistiwa sampai 90 derajat di kutub. *Longitude* atau garis bujur adalah digunakan untuk menentukan lokasi di wilayah barat atau timur dari garis utara selatan yang sering disebut juga garis *meridian*. Garis bujur diukur dari 0 derajat di wilayah Greenwich sampai 180 derajat di *International Date Line* seperti pada gambar 2.3.

Garis tengah diantara kutub merupakan sebuah lingkaran besar yang membagi bumi menjadi dua bagian yaitu utara dan selatan. Lingkaran garis imajiner ini akan semakin mengecil ketika mendekati

ti kutub. Garis lintang digunakan untuk membatasi corak iklim di permukaan bumi sedangkan garis bujur untuk menentukan waktu dan tanggal. Berikut ini pembagian iklim di bumi menurut bata garis lintang:

23,5-23,5 LU/LS	= iklim tropis
23,5-40 LU/LS	= iklim subtropis
40 LU-66,5 LU/LS	= iklim sedang
66,5 -90 LU/LS	= iklim kutub

Beda halnya dengan garis bujur atau *meridian*, ia tidak memiliki batas utama penanda awalnya suatu garis bujur sampai batas titik 0 derajat disepakati di Greenwich Inggris pada tahun 1884. Jadi sampai abad ke 19 tidak ada ketentuan baku untuk titik tolak penanggalan di bumi dan setiap wilayah/negara memiliki kebijakan masing-masing. Sama seperti garis lintang, jarak antar garis bujur juga ditulis dalam satuan derajat. Penulisan pada koordinat peta juga sama seperti penulisan untuk Garis Lintang. Yang membedakan hanyalah simbol huruf di belakangnya. Misalnya huruf B untuk Bujur Barat dan huruf T untuk Bujur Timur. Pada peta internasional, huruf E (*East*) untuk Bujur Timur dan huruf W (*West*) untuk Bujur Barat.



Latitude & Longitude

Gambar 2.3: *Latitude dan Longitude* [3]

2.6.4 Penentuan Posisi pada Android

Untuk menentukan posisi pengguna pada aplikasi android menggunakan *Location-based service*. Layanan tersebut bekerja dengan cara mencari lokasi sekarang dari perangkat yang digunakan. Aplikasi dapat meminta secara periodik pembaharuan informasi lokasi perangkat tersebut. Terdapat dua layanan *Location-based service* yang dapat digunakan aplikasi untuk mendapatkan lokasi perangkat, yaitu *Android Location Service* dan *Google Play Location Service*.

1. **Android Location Service**[9]

Android Location Service atau *Android Location Service API* memiliki banyak metode untuk mendapatkan informasi mengenai lokasi pengguna, berikut adalah dua metode yang digunakan dalam tugas akhir ini:

(a) **GPS_Provider**

Metode dari *Android Location Services* ini digunakan untuk menemukan lokasi pengguna menggunakan satelit. Faktor yang mempengaruhi layanan lokasi ini yaitu kondisi cuaca, letak geografis, dan ruangan tertutup. Metode ini membutuhkan waktu lebih lama dibandingkan layanan lokasi yang lain. Untuk Mengakses *GPS_PROVIDER* dibutuhkan izin *Fine Location* yaitu izin untuk mendapatkan lokasi yang akurat.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Kode 2.1: Izin menggunakan *Fine Location*

(b) **Network_Provider**

Berbeda dengan *GPS_PROVIDER*, *NETWORK_PROVIDER* mendapatkan lokasi pengguna dari ketersediaan pemancar telepon dan akses poin pada *WiFi* yang terhubung dengan perangkat pengguna. Layanan lokasi ini membutuhkan waktu yang lebih cepat untuk mendapatkan lokasi pengguna daripada *GPS_PROVIDER*, akan tetapi lokasi yang didapat tidak akurat. Layanan lokasi ini

membutuhkan izin untuk mengakses *Coarse location* yaitu izin untuk mendapatkan lokasi kasar pengguna.

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Kode 2.2: Izin menggunakan *Coarse Location*

2. Google Play Services Location

Layanan ini lebih unggul dari *Android Location Service* karena membuat aplikasi lebih akurat untuk mendapatkan lokasinya.

- (a) **Fused Location Provider** [10] Metode ini merupakan fitur dari *Google Play Service* yang menutupi kelemahan dari *API* layanan lokasi yang sebelumnya yaitu *API* akan menyesuaikan ketersediaan layanan lokasi. *API* ini bekerja dengan mengganti *provider* secara otomatis tergantung penggunaan energi, akurasi dengan cepat karena menggunakan *system-wide service*.

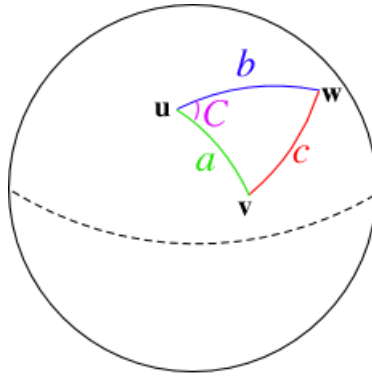
2.7 Formula *Haversine*

Formula *haversine* merupakan adalah persamaan penting dalam sistem navigasi, nantinya formula *haversine* ini akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (*longtitude*) dan garis lintang (*latitude*). *haversine* formula merupakan penerapan dari konsep trigonometri yang merupakan bagian dari geometri.. *Haversine* formula merupakan kasus khusus dari rumus yang lebih umum di trigonometri bola, hukum *haversines*, yang berkaitan dengan sisi dan sudut segitiga bola. Dari dua titik lingkaran(bumi) seperti pada gambar 2.4, *haversine* pada sudut tengah dari koordinat tersebut adalah [11] :

$$hav \frac{d}{r} = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)hav(\lambda_2 - \lambda_1) \quad (2.2)$$

Berikut perhitungan jarak dalam formula *haversine*:

$$d = 2r \arcsin(\sqrt{hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)hav(\lambda_2 - \lambda_1)}) \quad (2.3)$$



Gambar 2.4: Visualisasi hukum *Haversine*

Keterangan:

- hav = Nilai *haversine* pada sudut tengah antara dua koordinat
- d = Jarak antara dua koordinat
- r = radius bumi ($R = 6356.752$)
- φ_2 = *latitude* dari titik 2
- φ_1 = *latitude* dari titik 1
- λ_2 = *longitude* dari titik 2
- λ_1 = *longitude* dari titik 1

2.8 Google API

Google API adalah program yang dapat pengembang tambahkan pada aplikasi atau web untuk mengakses, menjalankan, memanfaatkan fungsi atau fitur yang disediakan *Google*.

2.8.1 Google Android API

1. *Map View*

sebuah *View* yang menampilkan peta dengan data yang diperoleh dari layanan *Google Maps*. Dapat memungkinkan pengguna untuk berinteraksi memindahkan peta dengan sentuhan.

2. *Marker*

Marker menunjukkan suatu lokasi dan menampilkan informasi yang berhubungan dengan lokasi pada peta. Bentuk, warna,

bahkan gambar khusus dapat diatur sendiri. Jendela info dapat memberikan konteks tambahan sebagai informasi.

3. **Polyline**

Polyline adalah serangkaian segmen garis terhubung yang bisa membuat bentuk apa saja yang diinginkan dan bisa digunakan untuk menandai jalur dan rute di peta.

2.8.2 Google Maps API

Adapun fitur-fitur *Google Maps API* yang digunakan pada penelitian ini adalah:

1. *Google Maps Android API* [12]

Menambahkan peta pada aplikasi *android* yang terintegrasi, bangunan 3D, rancangan dalam ruangan, *Street View* dan tampilan mode satelit, *custom marker*.

2. *Google Maps Directions API* [13]

Menemukan jalur antara dua lokasi atau lebih menggunakan serangkaian titik jalan. Untuk mengakses *API* tersebut menggunakan *URL* seperti pada gambar 2.5 . Hasil dari per-

```
https://maps.googleapis.com/maps/api/directions/json?  
origin=ORIGIN&destination=DESTINATION&key=YOUR_API_KEY
```

Gambar 2.5: *URL untuk akses Google Maps Directions API*

mintaan *API* lewat *URL* adalah data berupa *JSON*. Komponennya yaitu *JSON Array* dari *route* yang berisikan *legs*, didalam *JSON Array legs* berisikan *steps*. *JSON Array steps* merupakan kumpulan titik-titik hasil pencarian jalur yang diminta berisi *JSON Object start_location* dan *end_location*.

(a) **Routes**

Routes berisi satu hasil dari tempat asal dan tujuan yang ditetapkan. Rute ini bisa terdiri dari satu atau beberapa *legs* bergantung pada apakah ada titik jalan yang ditetapkan. Serta, rute tersebut juga berisi informasi hak cipta dan peringatan yang harus ditampilkan kepada pengguna di samping informasi rute.

(b) **Legs**

Legs menetapkan satu segmen perjalanan dari tempat

asal ke tujuan dalam rute yang dihitung. Untuk rute yang tidak berisi titik jalan, rute akan terdiri dari satu segmen, namun untuk rute yang mendefinisikan satu atau beberapa titik jalan, rute akan terdiri dari satu atau beberapa segmen, sesuai dengan segmen tertentu pada perjalanan tersebut.

(c) **Steps**

Steps mendefinisikan satu langkah dari arah yang dihitung. Langkah adalah satuan terkecil dari rute arah, berisi satu langkah yang menjelaskan satu petunjuk spesifik mengenai perjalanan. Tiap langkah memiliki *start_location* dan *end_location* sendiri.

i. *start_location*

berisi lokasi titik awal tiap langkah.

ii. *end_location*

berisi lokasi titik akhir tiap langkah.

2.9 Indeks Standar Pencemaran Udara

Saat ini Indeks standar kualitas udara yang dipergunakan secara resmi di Indonesia adalah Indeks Standar Pencemar Udara (IS-PU), hal ini sesuai dengan Keputusan Menteri Negara Lingkungan Hidup Nomor : KEP 45 / MENLH / 1997 Tentang Indeks Standar Pencemar Udara. Dalam keputusan tersebut yang dipergunakan sebagai bahan pertimbangan diantaranya : bahwa untuk memberikan kemudahan dari keseragaman informasi kualitas udara ambien kepada masyarakat di lokasi dan waktu tertentu serta sebagai bahan pertimbangan dalam melakukan upaya-upaya pengendalian pencemaran udara perlu disusun Indeks Standar Pencemar Udara. Indeks Standar Pencemar Udara adalah angka yang tidak mempunyai satuan yang menggambarkan kondisi kualitas udara ambien di lokasi dan waktu tertentu yang didasarkan kepada dampak terhadap kesehatan manusia, nilai estetika dan makhluk hidup lainnya. Indeks Standar Pencemar Udara ditetapkan dengan cara mengubah kadar pencemar udara yang terukur menjadi suatu angka yang tidak berdimensi. Rentang Indeks Standar Pencemar Udara dapat dilihat pada tabel 2.1.

Data Indeks Standar Pencemar Udara diperoleh dari pengo-

perasian Stasiun Pemantauan Kualitas Udara Ambien Otomatis. Sedangkan Parameter Indeks Standar Pencemar Udara meliputi :

1. Partikulat (PM_{10})
2. Karbondioksida (CO)
3. Sulfur dioksida (SO_2).
4. Nitrogen dioksida (NO_2).
5. Ozon (O_3)

Perhitungan dan pelaporan serta informasi Indeks Standar Pencemar Udara ditetapkan oleh Kepala Badan Pengendalian Dampak Lingkungan, yaitu Keputusan Kepala Badan Pengendalian Dampak Lingkungan No. 107 Tahun 1997 Tanggal 21 November 1997.2 Keputusan Kepala Badan Pengendalian Dampak Lingkungan, memuat diantaranya adalah Parameter-Parameter Dasar Untuk Indeks Standar Pencemar Udara (ISPU) Dan Periode Waktu Pengukuran, selengkapnya dapat dilihat pada tabel 2.2.

Tabel 2.1: Klasifikasi Warna Berdasarkan Indeks Kualitas Udara

ISPU	
Indeks	Kategori
1-50	Baik
51-100	Sedang
101-199	Tidak Sehat
200-299	Sangat Tidak Sehat
300 - lebih	Berbahaya

Tabel 2.2: Parameter-Parameter Dasar Untuk Indeks Standar Pencemar Udara (ISPU) Dan Periode Waktu Pengukuran

Parameter	Waktu Pengukuran
Partikulat (PM_{10})	24 jam (Periode pengukuran rata-rata)
Sulfur Dioksida (SO_2)	24 jam (Periode pengukuran rata-rata)
Carbon Monoksida(CO)	8 jam (Periode pengukuran rata-rata)
Ozon(O_3)	1 jam (Periode pengukuran rata-rata)
Nitrogen Dioksida(NO_2)	1 jam (Periode pengukuran rata-rata)

2.10 Akibat dari Kadar CO yang Berlebihan pada Tubuh

Akibat yang didapatkan apabila kadar CO dalam tubuh berlebihan yaitu keracunan gas CO, atau yang biasa disebut toksisitas. Gejala toksisitas atau keracunan ringan meliputi sakit kepala dan mual-mual pada konsentrasi kurang dari 100 ppm. Konsentrasi serendah 667 ppm dapat menyebabkan 50% hemoglobin tubuh berubah menjadi karboksihemoglobin (HbCO). Karboksihemoglobin cukup stabil, namun perubahan ini bisa reversibel atau dapat kembali ke keadaan awal. Karboksihemoglobin tidaklah efektif dalam menghantarkan oksigen di dalam sistem sirkulasi atau transportasi darah. Karena itu beberapa bagian tubuh tidak mendapatkan oksigen yang cukup. Sebagai akibatnya, paparan pada tingkat ini dapat membahayakan jiwa. Tabel 2.3 adalah data mengenai gejala yang ditimbulkan, selang waktu selama terpapar gas CO, dan konsentrasinya

Tabel 2.3: Data Gejala yang Didapatkan Selama Selang Waktu dan Konsentrasi Terpapar Gas CO [4]

Konsentrasi (ppm)	Waktu Terpapar	Gejala atau Efek
200	1-2 Jam	Sakit kepala dan gelisah
400	2-3 Jam	Sakit kepala dan merasa tidak enak badan serta gelisah
1000-2000	30 Menit	Palpitasi ringan pada jantung
	1 $\frac{1}{2}$ Jam	Kecenderungan terjadi sempoyongan
	2 Jam	Gangguan mental, sakit kepala, dan mual
2000-2500	30 Menit	Pingsan
>2500	Seketika	Terjatuh dan Meninggal

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Desain Sistem

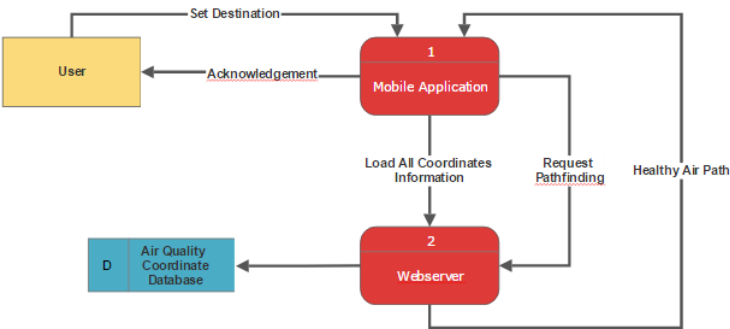
Penelitian ini bertujuan untuk membuat aplikasi navigasi berbasis *android* yang dapat menginformasikan pengguna dan menunjukkan jalur dengan kadar polusi terendah untuk meminimalisir penghirupan udara tidak sehat saat berolahraga serta beraktivitas diluar ruangan. Koordinat setiap persimpangan yang merupakan acuan untuk melakukan pencarian jalur dikumpulkan dengan menggunakan metode permintaan *API Google Maps Direction* secara rekursif ke seluruh daerah di Surabaya. Untuk penentuan titik awal digunakan metode *haversine* guna mendapatkan titik persimpangan terdekat dengan yang dipilih oleh pengguna. Dalam melakukan pencarian jalur pada tugas akhir ini, *cost* pada *graph* yaitu indeks kualitas udara pada jalan dikalikan dengan jaraknya. Berikut adalah sistem pada tugas akhir ini:

1. Pengumpulan koordinat persimpangan
2. Mendapatkan lokasi pengguna
3. Menentukan titik awal dan tujuan
4. Pencarian dan penggambaran jalur
5. Sistem navigasi

3.1.1 Data Flow Diagram

Data seluruh koordinat persimpangan jalan yang berada di *webserver* diakses oleh aplikasi untuk disajikan ke pengguna. pengguna kemudian memilih tujuan perjalanannya untuk di mintakan ke *webserver* oleh aplikasi pencarian jalur. Setelah jalur didapatkan ,aplikasi akan membaca data hasil pencarian jalur berupa jalur, jarak tempuh, serta total indeks. Data jalur digambarkan pada apli-

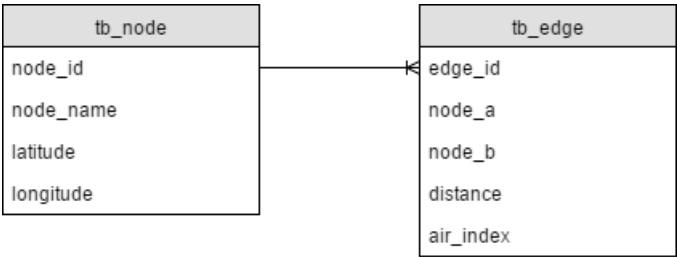
kasi sebagai informasi kepada pengguna seperti digambarkan pada gambar 3.1.



Gambar 3.1: Data Flow Diagram

3.1.2 Entity Relationship Diagram

Seperti digambarkan pada gambar 3.2 ,Struktur basis data dari tugas akhir ini terdiri dari dua tabel yaitu tabel *node* dan tabel *edge*. Tabel *node* berisi informasi tentang *node* seperti koordinat dan nama *node*. Tabel *edge* berisi informasi tentang indeks udara dan jarak antar *node*. Setiap anggota entitas tabel *node* dapat berhubungan dengan lebih dari satu anggota entitas tabel *edge* tetapi tidak sebaliknya. Data *node_a* memiliki hubungan dengan *tb_node* begitu juga *node_b*.

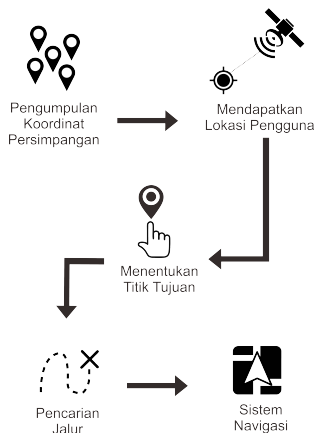


Gambar 3.2: Entity Relationship Diagram

3.2 Alur Kerja Sistem

Sistem Kerja berikut menggunakan *Android Studio* sebagai *toolkit* pembuatan aplikasi berbasis navigasi dan bahasa pemrograman *PHP* dalam melakukan pencarian jalur serta pengiriman *database*. Seperti pada gambar 3.3 Alur dari tugas akhir ini yaitu:

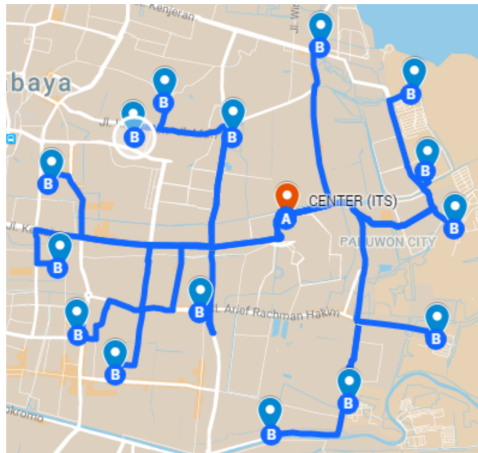
1. **Pengumpulan Koordinat Persimpangan**
Mendapatkan data dari seluruh koordinat persimpangan jalan di daerah Surabaya.
2. **Mendapatkan Lokasi Pengguna**
Mendapatkan lokasi pengguna menggunakan metode *fused location provider*.
3. **Menentukan Titik Awal dan Tujuan**
Fitur untuk memilih lokasi awal dan tujuan pengguna berdasarkan data koordinat persimpangan.
4. **Pencarian dan Penggambaran Jalur**
Metode untuk mendapatkan jalur dengan indeks kualitas udara terbaik dan menggambarkannya pada peta di aplikasi *android* dengan warna sesuai dengan ketentuan.
5. **Sistem Navigasi**
Sistem pemandu pengguna dalam melewati jalur yang telah ditampilkan setiap pengguna berpindah lokasi.



Gambar 3.3: Alur Kerja Sistem

3.3 Pengumpulan Koordinat Persimpangan

Pengumpulan koordinat persimpangan jalan bertujuan untuk mendapatkan data koordinat setiap persimpangan jalan besar yang ada di Surabaya untuk dijadikan graf acuan pencarian jalur. Pengumpulan koordinat yaitu dengan meminta dari *Google Maps Direction*, yaitu *API* untuk layanan pencarian jalur, dengan koordinat tengah wilayah Surabaya sebagai titik awal dan koordinat persimpangan jalan ujung wilayah Surabaya sebagai titik tujuan seperti gambar 3.4.



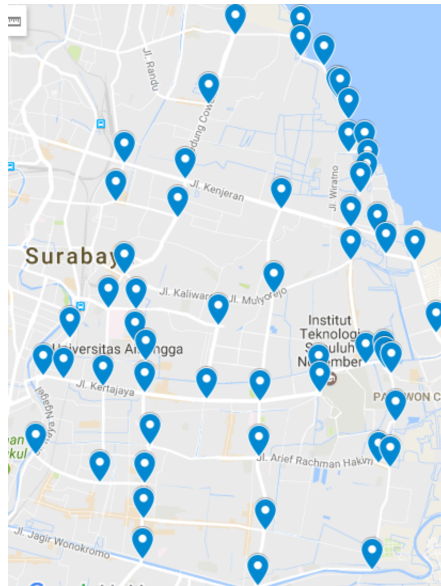
Gambar 3.4: Gambaran Koordinat Persimpangan Jalan yang Dikumpulkan

Hasil dari permintaan *API* adalah berupa jalur yang dilewati dari koordinat tengah Surabaya ke koordinat ujung wilayah Surabaya. Jalur tersebut berisikan kumpulan langkah-langkah, setiap langkah pada jalur merupakan titik koordinat persimpangan jalan dan jarak antar langkah, sehingga didapatkan beberapa koordinat persimpangan jalan dalam pencarian jalur lewat *API Google Maps Direction*. Permintaan *API* dilanjutkan secara rekursif hingga mengelilingi wilayah Surabaya seperti dijelaskan lewat *flowchart* pada gambar 3.6.

Respon dari permintaan *API* berisi *JSON Object* yang ber-

iskan banyak *JSON Array*. Langkah-langkah jalur yang dibutuhkan berada dalam *JSON Array* dengan nama *steps*. Dalam *JSON Array* *steps* yang akan dimasukkan ke basis data yaitu:

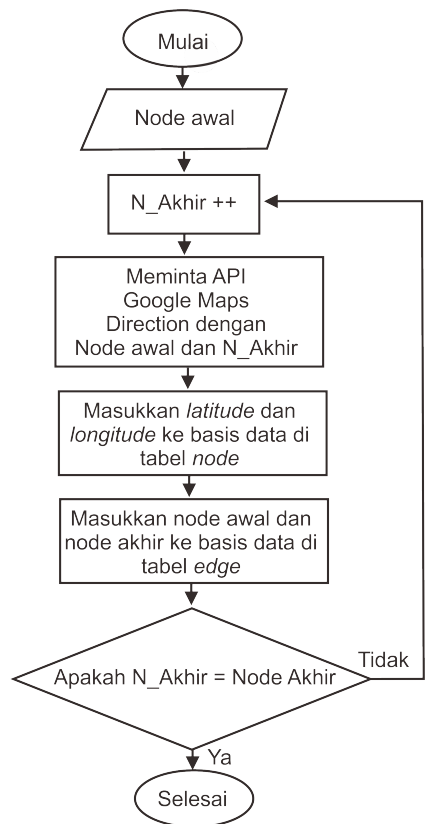
1. ***start_location***
berisikan informasi koordinat latitude dan longitude langkah awal.
2. ***end_location***
berisikan informasi koordinat *latitude* dan *longitude* langkah akhir, untuk langkah selanjutnya titik ini menjadi *start_location*.
3. ***distance***
berisikan informasi tentang jarak antara *start_location* dan *end_location*.



Gambar 3.5: Hasil Koordinat Persimpangan Jalan yang berhasil terkumpul

Setiap hasil dari permintaan *API* akan disimpan ke dalam basis data, sehingga didapatkan data setelah seluruh metode dijalankan seperti pada gambar 3.5 dengan daftar jumlah *node* dan *edge*

pada tabel 3.1 .



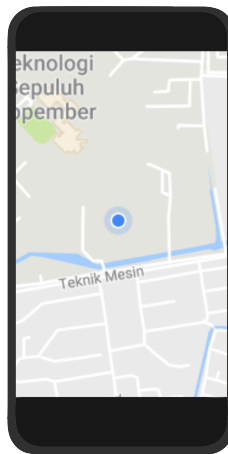
Gambar 3.6: Alur Pengumpulan Koordinat Persimpangan

Tabel 3.1: Koordinat yang dikumpulkan

Total <i>Node</i>	Total <i>Edge</i>
68 node	82 edge

3.4 Mendapatkan Lokasi Pengguna

Pada tugas akhir ini koordinat persimpangan jalan yang paling dekat dengan lokasi pengguna akan digunakan sebagai titik awal dalam pencarian jalur seperti pada gambar 3.7. Untuk mendapatkan lokasi pengguna pada *platform android* digunakan metode *Fused Location Provider*. Tujuan digunakannya metode *Fused Location*



Gambar 3.7: Mendapatkan lokasi pengguna

Provider adalah karena kelebihan dari metode tersebut yaitu dapat menyesuaikan *provider* yang dibutuhkan sesuai kondisi dan tidak perlu ada jeda menunggu lokasi. Dengan menggunakan prioritas *HIGH_ACCURACY* agar mendapatkan hasil maksimal dibandingkan prioritas yang lain tabel 3.2.

Tabel 3.2: Efek dari prioritas berbeda

Prioritas	<i>update interval</i>	Konsumsi Baterai	Akurasi
HIGH_ACURACY	5 detik	7.25%	10 meter
BALANCED_ACURACY	20 detik	0.6%	40 meter
NO-POWER	N/A	<i>small</i>	1 mil

3.5 Menentukan Titik Awal dan Tujuan

Koordinat lokasi pengguna dan koordinat yang dipilih oleh pengguna di peta kemudian dihitung untuk dicari lokasi koordinat persimpangan mana yang paling dekat dengan koordinat persimpangan jalan menggunakan formula *haversine* sehingga menjadi perwakilan lokasi pengguna dan lokasi tujuan pilihan pengguna seperti dijelaskan pada gambar 3.8

3.5.1 Perhitungan Jarak Antar Koordinat Menggunakan *Haversine Formula*

Perhitungan jarak antara koordinat yang dipilih oleh pengguna dengan koordinat persimpangan jalan pada data dilakukan secara keseluruhan memakai metode *brute force* seperti pada kode 3.1, satu persatu koordinat dihitung dari koordinat pertama hingga akhir, hasil perhitungan jarak dimasukkan kedalam array untuk dicari nilai jarak minimumnya.

```
DEKLARASI
procedure haversineformula {Menghitung jarak antara dua
    koordinat}
procedure indexminim {Mencari index dengan nilai terkecil}
procedure getvalue {Mendapatkan nilai dalam ArrayList p[ada
    index tertentu}

nodekoordinat : ArrayList {List koordinat persimpangan
    jalan}
koordinatsekarang: LatLng {Koordinat lokasi pengguna}
jumlahkoordinat : Int {Jumlah total koordinat persimpangan
    jalan}
Double hasiljarak[kumpulan jarak perhitungan haversine]

DESKRIPSI
for i=0 to jumlahkoordinat do
    hasiljarak[i] = haversineformula(nodekoordinat,
        koordinatsekarang)
endfor
```

Kode 3.1: Mencari koordinat persimpangan jalan yang paling dekat dengan suatu koordinat menggunakan formula haversine untuk klasifikasi

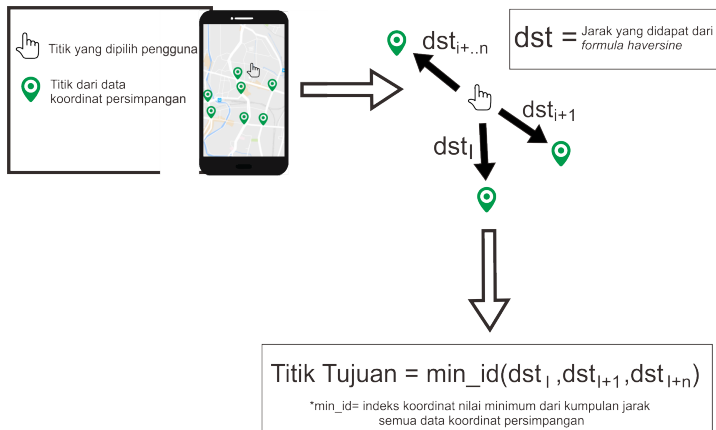
3.5.2 Mencari Indeks Koordinat Terdekat

Kumpulan perhitungan jarak antar koordinat berupa *array* kemudian dicari indeks dengan nilai jarak minimum dalam *array* seperti dijelaskan pada kode 3.3. Indeks yang didapatkan merupakan indeks koordinat yang akan dipakai sebagai koordinat tujuan.

```
DEKLARASI
Double hasiljarak[kumpulan jarak perhitungan haversine]
temp: double{nilai indeks minimal temporer dimulai dari
    indeks pertama array}
min_id: int{menyimpan indeks minimal dari temp}

DESKRIPSI
temp = hasiljarak[0]
min_id=0
for i = 1 to do
if hasiljarak[i] < temp:
temp = hasiljarak[i]
min_id = i
endfor
```

Kode 3.2: Mencari indeks koordinat persimpangan paling dekat dengan koordinat pilihan pengguna



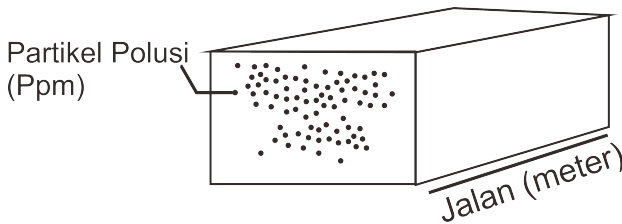
Gambar 3.8: Alur penentuan titik tujuan

3.6 Pencarian dan Penggambaran Jalur

Tahap pencarian jalur bertujuan untuk mendapatkan jalur yang memiliki kadar udara yang baik. pencarian jalur dilakukan menggunakan algoritma *dijkstra*. Tahap ini merupakan tahapan penting karena inti dari tugas akhir ini bertumpu pada tahap ini. Tahap setelah pencarian jalur yaitu penggambaran jalur pada aplikasi yang bertujuan untuk menginformasikan kepada pengguna tentang jalur dengan kualitas udara terbaik dari titik awal ke titik akhir sesuai permintaan pengguna.

3.6.1 Bobot *Graph*

Pada tugas akhir ini jalur yang diharapkan adalah jalur dengan kualitas udara terbaik. Sehingga indeks kualitas udara setiap jalan yang sudah ada menjadi acuan dalam pencarian jalur.



Gambar 3.9: Gambaran Polusi Udara pada Jalan

Polusi udara berada pada dimensi ruang yang berjarak bila di gambarkan akan pada gambar 3.9 sehingga setiap satuan jarak terdapat polusi. Karena tugas akhir ini bertujuan untuk meminimalisir penghirupan polusi, maka diberikan prioritas setiap kategori indeks polusi seperti pada tabel 3.3 berupa konstanta seperti pada perhitungan berikut:

$$E = P \times I \times d \quad (3.1)$$

Keterangan:

Tabel 3.3: Konstanta Prioritas Indeks Kualitas Udara

Kategori Indeks Kualitas Udara	Konstanta Prioritas	Kategori
1-50	1	Baik
51-100	10	Sedang
101-199	30	Buruk
200-299	40	Sangat Buruk
300>	100	Berbahaya

- E = bobot graf
 P = konstanta prioritas indeks kualitas udara
 I = nilai indeks kualitas udara pada jalan
 d = jarak pada jalan

Hasil dari perkalian indeks dengan jarak menghasilkan bobot yang diinginkan. Perkalian tersebut bertujuan agar indeks kualitas udara dan jarak jalan dipertimbangkan dalam pencarian jalur. *Graph* setelah diubah bobotnya dapat divisualisasikan seperti gambar 3.10.

```
{"1":{"2":69650,"4":26350},"2":{"1":69650,"3":66000},"3":{"2":66000,"5":  
{"3":258000,"6":564300,"7":228900,"15":146900},"6":{"5":564300,"18":17  
{"8":93100,"14":300800},"10":{"7":68800,"8":215000,"13":144500},"11":{"  
{"9":300800,"15":259000},"15":{"14":259000,"5":146900,"16":195800},"16  
{"3":114000,"20":116600,"40":45500},"20":{"19":116600,"6":187000,"25":7  
{"22":403200,"25":104650},"25":{"23":104650,"24":104250,"20":74250,"21  
{"27":122250,"29":141000,"33":235300,"55":126750},"29":{"28":141000,"3  
{"31":172200,"27":211500,"3":68400,"33":324800},"33":{"32":324800,"28"  
{"25":256200,"35":186200,"37":90500},"35":{"36":186200,"41":183000},"3  
{"37":184500,"19":45500},"41":{"34":74200,"35":183000},"42":{"45":3480  
{"45":15050,"47":14850},"47":{"46":14850,"48":21000,"49":3750},"48":{"4  
{"51":11760,"53":40950},"53":{"52":40950,"54":37680},"54":{"53":37680,"
```

Gambar 3.10: Data *Graph* Jalan yang Telah Dikumpulkan Dipengaruhi oleh Jarak dan Kualitas Udara

3.6.2 Pencarian Jalur Menggunakan Algoritma *Dijkstra*

```
DEKLARASI
graf := int[graf dari data koordinat persimpangan dengan
        bobot yang sudah disesuaikan]
titikawal := int{indeks dari titik awal pencarian jalur}
node := int{indeks titik koordinat}
v := vertex in Q with min dist[v]
jalur :

DESKRIPSI
function Dijkstra(graf, titikawal):
    jalur[titikawal] = 0
    for each node in graf:
        if node is not titikawal
            jalur[node] := takhingga
            add v to Q

    while Q is not empty:
        for each titikakhir of node:
            alt := jalur[node] + length(node,titikawal)
            if alt < jalur[node]:
                jalur[u] := alt

    return jalur[]
end function
```

Kode 3.3: Algoritma Dijkstra untuk Pencarian Jalur

Tujuan penggunaan Algoritma *Dijkstra* dalam pencarian jalur yaitu untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. *Graph* yang dibuat sebelum dimasukkan data persimpangan jalan diberi nilai bobot untuk setiap titik ke titik lainnya dengan nilai 0 pada titik awal dan nilai tak hingga terhadap titik lain yang belum terisi nilai. Tandai seluruh *node* dengan penanda belum di periksa dan tandai titik pertama sebagai titik awal. Dari titik awal, pertimbangkan titik tetangga yang belum diperiksa dan hitung jaraknya dari titik awal. Ketika selesai memeriksa setiap jarak terhadap titik tetangga, tandai titik yang telah terjamah sebagai titik yang telah diperiksa. Titik yang telah diperiksa tidak akan pernah di periksa kembali, jarak yang disimp-

an adalah jarak terakhir dan yang paling minimal bobotnya. tandai titik yang belum diperiksa dengan jarak terkecil dari titik awal sebagai titik awal selanjutnya. Hasil dari pencarian jalur dapat dilihat pada gambar 3.11

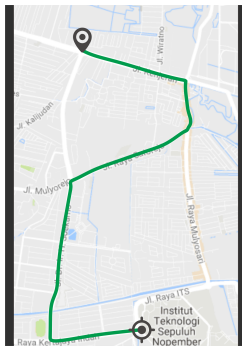
```
"{
  "jalur" : {"0":"1","1":2,"2":"3"},
  "index" : {"0":"199"},
  "jarak" : {"0":"350"},
  "latitude" : {"0":"-7.276407","1":"-7.279156","2":"-7.280585"},
  "longitude" : {"0":"112.79036","1":"112.790794","2":"112.780819"}
}"
```

Gambar 3.11: Hasil Pencarian jalur menggunakan bahasa *PHP* dari data kualitas udara

3.6.3 Penggambaran dan Pewarnaan Jalur Pada Aplikasi

Dari jalur yang didapat melalui permintaan aplikasi ke *webserver* lalu aplikasi memvisualisasikannya pada peta. Penggambaran jalur pada peta dari titik awal dan titik tujuan yang merupakan data koordinat persimpangan jalan ditambahkan titik awal dari jalur yaitu titik lokasi pengguna pada awal langkah jalur. Contoh visualisasi jalur pada peta dapat dilihat pada gambar 3.12(a). Bobot indeks kualitas udara dan jarak yang menentukan jalur yang akan ditampilkan. Karena telah ditentukan pengaturan bobot *default* jalan apabila tidak ada data dari alat pengumpul indeks kualitas udara maka pewarnaan jalur berwarna hijau.

Jalur tersebut lalu diwarnai berdasarkan ketentuan dari indeks standar pencemaran udara. Warna hijau sebagai indikasi bahwa jalan tersebut memiliki indikasi udara baik, warna biru mengindikasikan jalan tersebut sedang, warna kuning mengindikasikan tidak sehat, warna merah mengindikasikan sangat tidak sehat, dan warna hitam mengindikasikan berbahaya. apabila terdapat data dari alat pengumpul indeks kualitas udara seperti digambarkan pada gambar 3.12(b).



(a) Contoh Hasil Penggambaran Jalur

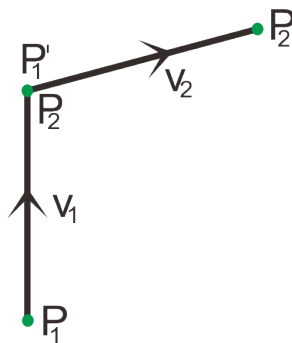


(b) Contoh Hasil Pewarnaan Jalur

Gambar 3.12: Contoh Penggambaran dan Pewarnaan Jalur

3.6.4 Sistem Navigasi

Karena sasaran pengguna aplikasi ini adalah pengguna yang bersepeda dan berlari maka harus ditambahkan fitur non-visual untuk memandu pengguna melewati jalur yang sudah ada. Fitur ini akan memandu dengan mengeluarkan petunjuk berupa suara dari aplikasi yang mengisyaratkan pengguna untuk "belok kanan" atau "belok kiri".



Gambar 3.13: Penggambaran dan Pewarnaan Jalur

Sistem akan mendeteksi persimpangan selanjutnya akan berada di kiri atau di kanan jalan dengan menghitung perubahan vektor dari jalan pertama ke jalan kedua. Perhitungan dilakukan dari awal jalur didapatkan, sehingga didapatkan indikasi arah persimpangan selanjutnya. Untuk perintah suara dan pembaruan pencarian jalur dan penggambaran jalur akan dikeluarkan 10 meter sebelum persimpangan Seperti digambarkan pada gambar 3.13 dengan menggunakan rumus 3.2. Pembaruan pencarian dan penggambaran jalur dilakukan untuk mengantisipasi adanya perubahan data selama navigasi berlangsung.

$$\begin{aligned}
 V_1 &= P_2 - P_1 \\
 V_2 &= P_2' - P_1' \\
 V_1 &= (X_1, Y_1) \\
 V_2 &= (X_2, Y_2) \\
 Z &= (X_2 \times Y_1) - (X_1 \times Y_2)
 \end{aligned} \tag{3.2}$$

Keterangan:

P_1	=	Titik awal pada jalur pertama
P_2	=	Titik akhir pada jalur pertama
P_1'	=	Titik awal pada jalur kedua
P_2'	=	Titik akhir pada jalur kedua
V_1	=	Vektor jalan pertama berisikan (X_1, Y_1)
V_2	=	Vektor jalan kedua berisikan (X_2, Y_2)
Z	=	Nilai Perubahan Vektor

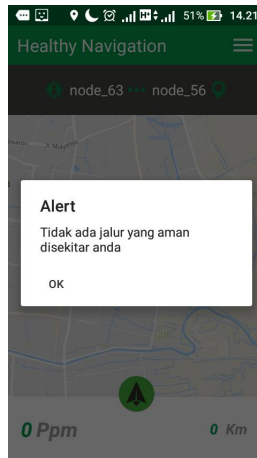
Apabila nilai hasil perubahan vektor jalan dari jalan pertama menuju jalan kedua lebih dari nol, maka mengindikasikan arah persimpangan tersebut menuju ke kanan kemudian aplikasi pun akan mengeluarkan perintah suara "belok kanan" sebagai isyarat kepada pengguna untuk berbelok ke kanan. Begitu juga sebaliknya apabila nilai hasil perubahan vektor dari jalan pertama ke jalan kedua kurang dari nol, maka nilai tersebut merupakan indikasi arah persimpangan tersebut menuju ke kiri dan aplikasi pun akan mengeluarkan perintah suara "belok kiri". Penentuan arah dapat dilihat pada tabel 3.4.

Tabel 3.4: Penentuan Arah Persimpangan

Perubahan Vektor	Perintah Arah
> 0	<i>Turn Right</i>
< 0	<i>Turn Left</i>
$= 0$	<i>Go Straight</i>

3.6.5 Kondisi Saat Seluruh Jalan Berbahaya

Apabila saat pengguna ingin melakukan pencarian jalur dan pada saat itu juga kondisi seluruh jalan pada *database* bernilai buruk yang dapat membahayakan pengguna apabila melewati jalur tersebut maka aplikasi akan memunculkan dialog peringatan kepada pengguna bahwa kondisi udara daerah tersebut berbahaya untuk kesehatan selama berolahraga dan sistem tidak melanjutkan visualisasi pencarian jalur. Seperti pada gambar 3.14.

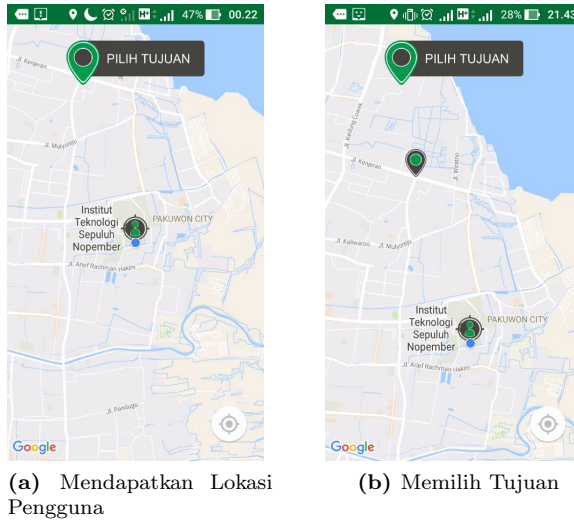


Gambar 3.14: Dialog Peringatan Tentang Kondisi Jalanan yang Buruk

3.7 Implementasi Antarmuka Aplikasi

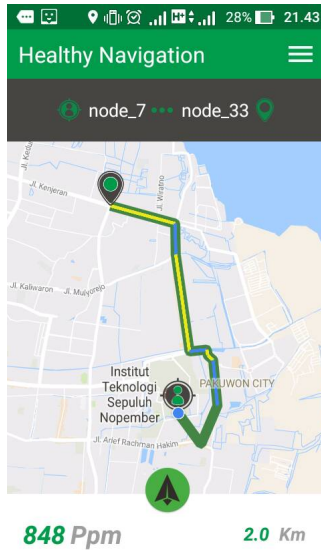
Aplikasi ini dirancang untuk perangkat *android* dengan minimum versi *SDK* 18 (*Jelly Bean*) dan target versi *SDK* 25 (*Lolipop*). Berikut adalah contoh tampilan antarmuka aplikasi navigasi berdasarkan kualitas udara:

3.7.1 Antarmuka Pemilihan Lokasi Awal dan Tujuan



Gambar 3.15: Antarmuka Menampilkan Lokasi Pengguna dan Memilih Tujuan

Pada antarmuka seperti pada gambar 3.15(a) dan (b) aplikasi akan mendapatkan lokasi pengguna dengan menggunakan layanan *Fused-Location Services* kemudian menampilkannya dengan marker untuk menginformasikan pada pengguna. Terdapat enam objek pada tampilan peta yang merupakan fitur dari *Google Maps*, Tombol *current location* untuk mendapatkan lokasi pengguna serta memindahkan kamera peta ke lokasi pengguna di tengah-tengah layar, Marker *Your Location* untuk merepresentasikan koordinat pengguna, Marker *Destination* untuk merepresentasikan koordinat tujuan, dan Tombol Pilih Tujuan untuk memverifikasi bahwa tujuan yang dipilih sudah benar lalu melanjutkan ke aktivitas selanjutnya yaitu pencarian jalur.



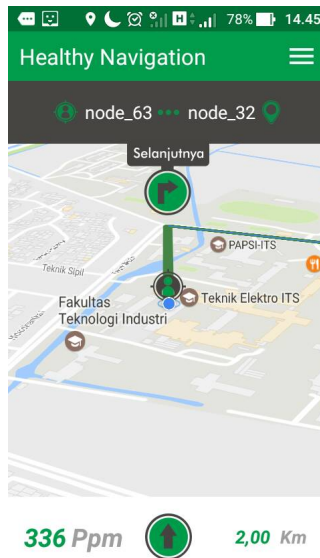
Gambar 3.16: Penggambaran dan Pewarnaan Jalur

3.7.2 Antarmuka Penggambaran dan Pewarnaan Jalur

Setelah hasil pencarian jalur didapatkan kemudian aplikasi menggambar jalur tersebut pada peta jalur akan digambarkan menggunakan *polyline* berwarna hijau sebagai latar belakang jalur. Kategori nilai indeks berdasarkan kategori indeks standar pencemaran udara. Pewarnaan jalur disesuaikan seperti pada tabel 3.5. Alamat lokasi awal dan tujuan ditampilkan pada layout atas. Total kadar polusi CO dan jarak yang akan ditempuh pada jalur ditampilkan pada *TextView* yang berada di *layout* bawah, bertujuan untuk menginformasikan kepada pengguna bahwa total kadar udara yang akan dilewati adalah yang paling baik. Tombol *Start Navigation* untuk melakukan perjalanan sehingga aplikasi akan memasuki mode navigasi.

Tabel 3.5: Klasifikasi Warna Jalur Berdasarkan Kategori ISPU

Indeks	Warna	Kategori
1-50	Hijau	Baik
51-100	Biru	Sedang
101-199	Kuning	Tidak Sehat
200-299	Merah	Sangat Tidak Sehat
300-lebih	Hitam	Berbahaya



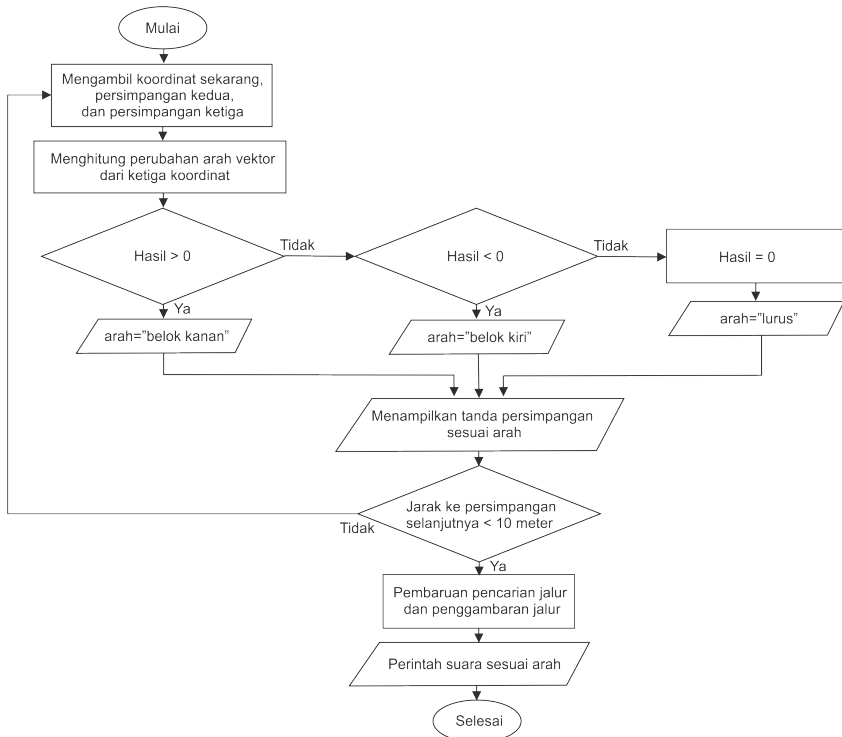
Gambar 3.17: Antarmuka Sistem Navigasi

3.7.3 Antarmuka Sistem Navigasi

Dalam mode ini kamera akan menghadap ke arah jalur pertama menghadap dan memiringkan kamera 30 derajat kebawah. Pengguna akan *ditracking* pergerakannya agar tidak keluar dan salah jalur. Pembaruan visualisasi dilakukan apabila pengguna berpindah lokasi dengan selang pembaruannya selama 1 detik. Setelah pembaruan dilakukan jalur yang terhubung dengan lokasi pengguna juga akan diperbarui sehingga akan terus menempel dengan lokasi

pengguna.

Terdapat dua penanda arah navigasi. Penanda yang berada di *layout* bawah adalah panduan arah yang dilakukan pengguna sekarang, sedangkan penanda yang berada di bawah layout atas adalah penanda arah pada persimpangan selanjutnya, apakah jalur akan belok ke kanan atau ke kiri.



Gambar 3.18: *Flowchart* Navigasi Arah Persimpangan dan Suara

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini dilakukan pengujian fitur-fitur pada aplikasi navigasi berdasarkan kualitas udara.

4.1 Metode Pengujian

Pada penelitian ini, pengujian dilakukan menggunakan lima aspek yang berbeda-beda. Aspek yang diuji adalah:

1. Perbandingan Metode Layanan Lokasi

Membandingkan layanan untuk menemukan lokasi perangkat pengguna dengan tiga layanan lokasi, yaitu:

(a) *GPS Provider*

Dalam mendapatkan lokasi pengguna menggunakan layanan dari *Android Location Service*, dengan memanfaatkan satelit untuk mendapatkan lokasi.

(b) *Network Provider*

Layanan *Location Service* ini menemukan lokasi perangkat yaitu dari ketersediaan pemancar telepon dan akses poin pada *WiFi* yang terhubung dengan perangkat pengguna

(c) *Fused Location Provider*

Layanan dari *Google Play Location Service* ini mendapatkan lokasi perangkat dengan mengganti *provider* secara otomatis tergantung penggunaan energi, akurasi dengan cepat

2. Pencarian Jalur

(a) Total Jarak yang Didapatkan

Pencarian jalur dengan dua kasus, yaitu kasus dimana bobot jalan pencarian jalur hanya jarak saja dan kasus dimana bobot jalan pencarian jalur berisi acuan indeks kualitas udara, konstanta prioritasnya, serta jarak jalan.

(b) Total Indeks Kualitas Udara yang Didapatkan

Pencarian jalur dengan membandingkan jumlah indeks kualitas udara yang didapatkan dari pencarian jalur menggunakan sistem *Google Maps* dan sistem tugas akhir.

3. Penentuan Arah Pada Persimpangan

Dalam sistem navigasi pengguna akan diarahkan menggunakan suara. Pengujian ini bertujuan untuk melihat akurasi kebenaran metode penentuan arah.

4. Waktu Eksekusi Setiap Alur Aplikasi

Aplikasi ini memiliki fitur-fitur setiap alurnya, pengujian ini menghitung berapa lama setiap fitur alur diproses.

5. Uji Ketergunaan Aplikasi

Aplikasi ini bertujuan untuk dapat menginformasikan pengguna dan memandunya pada jalur dengan kualitas udara terbaik, maka harus dilakukan pengujian ketergunaan aplikasi untuk membuat aplikasi ini layak untuk dipakai khalayak umum.

4.2 Perbandingan Metode Layanan Lokasi

Metode perhitungan lokasi perangkat pengguna akan di uji dengan menggunakan tiga metode yang umum digunakan untuk mendapatkan lokasi. Tujuan dari pengujian ini adalah menentukan metode terbaik yang ditentukan oleh selisih jarak terkecil dari koordinat yang didapatkan oleh *Google Maps* dengan koordinat yang didapatkan oleh masing-masing metode.

4.2.1 GPS Provider

Pengujian dilakukan dengan mendapatkan lokasi dengan menggunakan *Google Maps* dan metode layanan lokasi pada aplikasi di tempat yang sama sebanyak sembilan kali, didapatkan masing-masing selisih jarak antara kedua koordinat yang diuji. Hasil Pengujian Perbandingan dapat dilihat pada tabel 4.1

Pada pengujian antara *Google Maps* dengan layanan lokasi *GPS Provider* didapatkan rata-rata 6.294 Meter selisih antara dua koordinat yang didapatkan dari dua metode pengujian. Selisih jarak koordinat yang didapat terbesar adalah 11.678 dan yang terkecil adalah 0.560, yang artinya koordinat yang didapat oleh layanan lokasi ini cukup akurat. Kelemahannya adalah membutuhkan waktu yang lebih untuk mendapatkan lokasi hal ini dapat disebabkan oleh kondisi cuaca maupun geografis.

4.2.2 Network Provider

Hasil lokasi yang didapatkan oleh *Network Provider* jika dibandingkan dengan hasil dari *Google Maps*, selisih antara kedua

Tabel 4.1: Perbandingan dengan Menggunakan *GPS Provider*

<i>Google Maps</i>	<i>Android</i>	Selisih Jarak(meter)
-7.281992, 112.788599	-7.2819666666, 112.7886249999	4.019
-7.280523, 112.78888	-7.28053000001, 112.7888888	1.244
-7.279939, 112.791387	-7.279943333, 112.79146666	8.799
-7.280270, 112.792161	-7.280203333, 112.79217500001	7.572
-7.281249, 112.792761	-7.281223333, 112.7926583333	11.678
-7.282275, 112.792546	-7.28227, 112.792546666	0.560
-7.283116, 112.792694	-7.283113333, 112.792660000	3.761
-7.286039, 112.793543	-7.28601666, 112.7936116666	7.970
-7.285229, 112.795864	-7.285135, 112.7958183333	11.602
Rata-rata		6.294

hasil sangat jauh. Hasil selisih dari perbandingan tersebut apabila dibandingkan dengan hasil selisih dari *Google Maps* dan *GPS Provider* dapat diamati bahwa akurasi layanan lokasi ini memiliki akurasi yang jelek. Pada tabel 4.2 tersusun hasil selisih koordinat yang didapatkan dari *Google Maps* dengan layanan lokasi *Network Provider*.

Tabel 4.2: Perbandingan dengan Menggunakan *Network Provider*

<i>Google Maps</i>	<i>Android</i>	Selisih Jarak(Meter)
-7.282102, 112.788139	-7.280205, 112.787376	227.105
-7.280255, 112.789338	-7.280259, 112.7894173	8.757
-7.279900, 112.787067	-7.2793916, 112.7864552	88.030
-7.276830, 112.786218	-7.2778753, 112.7816004	522.412
-7.275490, 112.791616	-7.275229, 112.7916052	29.046
-7.274365, 112.797088	-7.245622, 112.7969814	3196.097
-7.279950, 112.791442	-7.2781081, 112.7920744	216.362
-7.282195, 112.792435	-7.2858101, 112.7937465	427.216
-7.285169, 112.795582	-7.2852152, 112.7960741	54.519
Rata-rata		529.949

Dari tabel diatas dapat dilihat bahwa rata-rata selisih koordinat yang didapatkan antara *Google Maps* dengan layanan lokasi NETWORK_PROVIDER adalah 529,949 Meter. Nilai selisih perbandingan lokasi yang cukup besar tersebut mengindikasikan bahwa akurasi dari layanan lokasi ini buruk. Tetapi waktu yang dibutuhkan untuk mendapatkan lokasi tidak banyak apabila perangkat ter-

sambung melalui jaringan *WiFi* atau jaringan telepon seluler.

4.2.3 *Fused Location Provider*

Jika diamati hasil dari lokasi yang didapatkan oleh layanan lokasi ini apabila dihitung selisihnya dengan koordinat yang diperoleh dari *Google Maps* lebih kecil jika dibandingkan dengan selisih dengan *Google Maps* dari layanan lokasi *GPS Provider* dan *Network Provider*. Pada tabel 4.3 dapat dilihat susunan hasil lokasi yang didapatkan dari *Google Maps* dan *Fused Location Provider* dengan selisih antara keduanya.

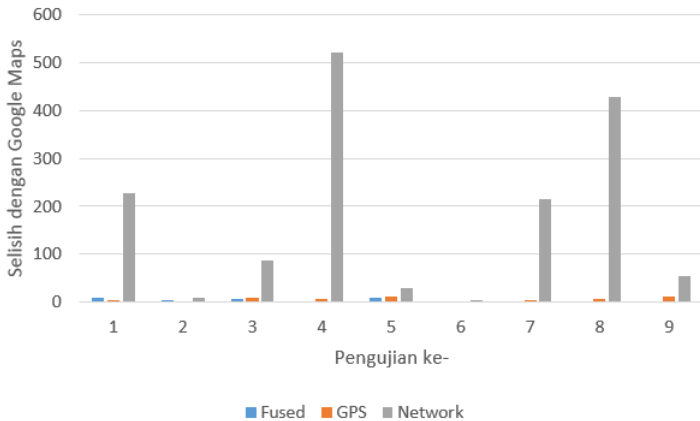
Tabel 4.3: Perbandingan dengan Menggunakan *Fused Location Provider*

<i>Google Maps</i>	<i>Android</i>	Selisih Jarak (Meter)
-7.285098, 112.796294	-7.2851333, 112.796367	8.957
-7.282025, 112.785827	-7.282055, 112.7858483	4.080
-7.281994, 112.788555	-7.2819617, 112.7886133	7.365
-7.281262, 112.787454	-7.2812567, 112.7874767	2.572
-7.279671, 112.790898	-7.27975, 112.790915	8.982
-7.282073, 112.792551	-7.28208, 112.7925667	1.898
-7.2834233, 112.7932317	-7.283408, 112.793227	1.778
-7.286244, 112.793789	-7.2862533, 112.7937933	1.137
-7.285741, 112.796814	-7.2857367, 112.7968083	0.789
Rata-rata		4.086

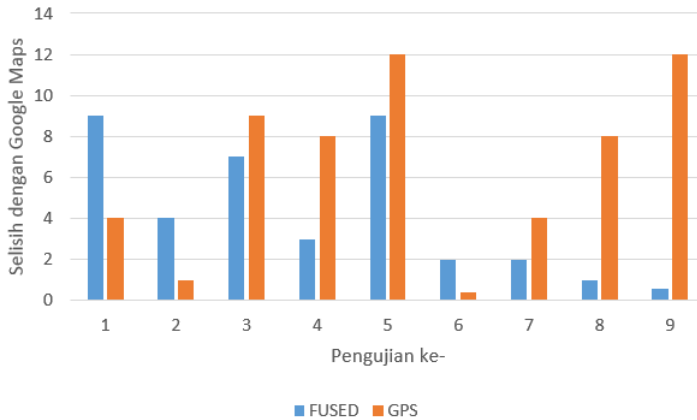
Pada tabel diatas dapat ditemukan nilai selisih 0,789 Meter yang merupakan selisih paling kecil dalam pengujian pada layanan lokasi ini. Nilai tersebut jika dibandingkan dengan hasil selisih terkecil dari *GPS Provider* yaitu 0,560 Meter memang kalah, akan tetapi rata-rata selisih yang didapatkan dari pengujian perbandingan ini adalah 4,086 Meter. Nilai tersebut jika dibandingkan dengan hasil dari *GPS Provider* lebih unggul. Selisih paling besar dari pengujian ini juga lebih unggul dengan nilai selisih terbesar yaitu 8,982 Meter.

Dari grafik gambar perbandingan ketiga layanan lokasi pada gambar 4.1 dapat dilihat bahwa *Network Provider* merupakan layanan lokasi yang selisih koordinatnya paling besar dan tidak konstan nilainya yang digambarkan dengan bar warna biru, jika dibandingkan dengan kedua layanan lokasi lainnya juga perbandingannya sangat jauh sehingga dapat diketahui bahwa akurasi *Network Provider* sangat jelek.

Jika *Network Provider* dipisah dan hanya meninggalkan dua layanan lokasi yang tersisa seperti grafik gambar 4.2, dapat diamati bahwa selisih perbandingan keakurasian kedua layanan lokasi tidak berbeda tidak terlalu jauh dengan *GPS Provider* digambarkan dengan bar berwarna oranye dan *Fused Location Provider* digambarkan dengan bar berwarna abu-abu. Akan tetapi nilai selisih dari Fused Location adalah yang paling banyak menjadi nilai minimum dari perbandingan kedua layanan lokasi tersebut, artinya *Fused Location Provider* memiliki keakurasian yang lebih dari ketiga layanan lokasi lain.



Gambar 4.1: Perbandingan Ketiga Selisih Layanan Lokasi



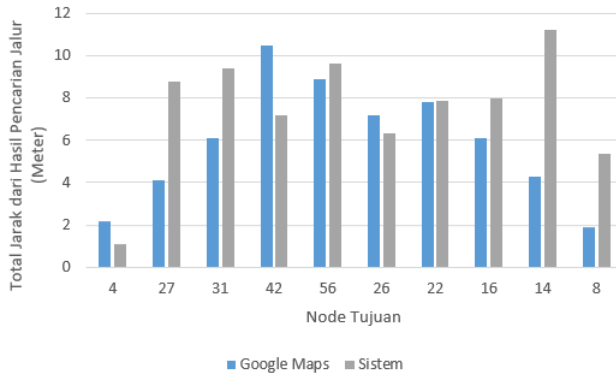
Gambar 4.2: Perbandingan Selisih *GPS* dan *Fused Location*

4.3 Pencarian Jalur

Pencarian jalur yang dilakukan oleh sistem tugas akhir ini memakai indeks kualitas udara sebagai bobot tambahan sebagai acuan pencarian jalur dengan ditambahkan aspek prioritas indeks kualitas jalan. Pada pengujian ini akan diuji dua kasus, kasus pertama adalah membandingkan jarak yang didapatkan apabila bobot yang dipakai hanya jarak saja atau ditambahkan aspek indeks kualitas udara beserta prioritasnya. Kasus yang kedua hasil pencarian jalur dari sistem tugas akhir ini akan dibandingkan dengan hasil dari pencarian jalur dari *Google Maps* dengan koordinat awal dan tujuan yang sama, dihitung total indeks kualitas udara yang didapatkan menggunakan dua sistem tersebut.

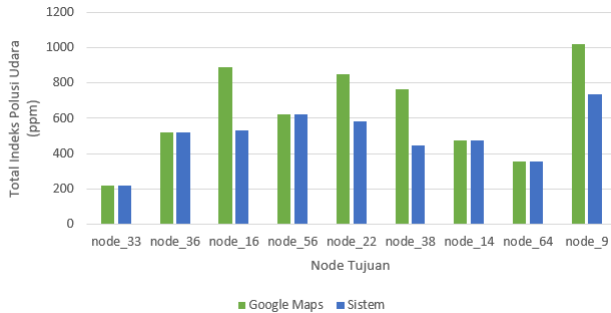
4.3.1 Total Jarak yang Didapatkan

Jika diamati dari grafik gambar 4.3, nilai jumlah jarak yang didapatkan dari pencarian jalur apabila bobot jalan hanya jarak jalan saja rata-rata lebih dekat daripada jumlah jarak yang didapatkan dari hasil pencarian jalur yang berbobot jarak dengan acuan lain berupa indeks kualitas udara dan prioritasnya. Hal tersebut dikarenakan tujuan dari bobot tipe kedua adalah untuk mencari jalur dengan indeks kualitas udara yang lebih baik daripada pasangan jalan tujuan lainnya, meskipun jarak yang ditempuh lebih jauh.



Gambar 4.3: Perbandingan Total Jarak yang Didapatkan dari Dua Tipe Bobot

4.3.2 Total Indeks Kualitas Udara yang Didapatkan



Gambar 4.4: Perbandingan Total Indeks Kualitas Udara yang Didapatkan dari Sistem Tugas Akhir dan Sistem Tugas Akhir

Dari gambar grafik 4.4 jumlah nilai indeks kualitas udara yang didapatkan dari sistem pencarian jalur *Google Maps* rata-rata lebih buruk daripada jumlah nilai indeks kualitas udara yang didapatkan dari sistem pencarian jalur menggunakan sistem tugas akhir. Di-karenakan bobot dari jalur pada sistem tugas ditambahkan aspek indeks kualitas udara dan prioritasnya, sehingga pemilihan jalur

akan memprioritaskan indeks kualitas udara yang lebih baik. Hasil tersebut sesuai dengan tujuan dari tugas akhir ini yaitu memprioritaskan jalur yang indeks kualitas udaranya lebih baik agar mencegah paparan gas CO yang berlebihan pada pengguna.

4.4 Penentuan Arah Pada Persimpangan

Tabel 4.4: Pengujian Penentuan Arah Pada Persimpangan

Pengujian ke-	Hasil	Keluaran Seharusnya	Keluaran Pengujian
1	4,269680000003247E-5	Kanan	Kanan
2	-3,408479999996993E-5	Kiri	Kiri
3	-1,768890000000165E-4	Kiri	kiri
4	-0.11786620669991654	Kiri	Kiri
5	0.2654151650000604	Kanan	Kanan
6	8.873559999998654E-5	Kanan	Kanan
7	1.2163800000002416E-5	Kanan	Kanan
8	1.8085199999993547E-5	Kanan	Kanan
9	-4.934229999994745E-5	Kiri	Kiri
10	-5.7734999999950874E-5	Kiri	Kiri

Pengujian penentuan arah pada persimpangan jalan ini dilakukan sebanyak sepuluh kali dengan cara mendapatkan nilai dari tiga koordinat persimpangan yang dilalui pengguna kemudian diberikan keluaran. Hasil keluaran lalu dibandingkan dengan keluaran yang seharusnya keluar dari keadaan asli. Dapat diamati pada tabel 4.4 mengenai hasil pengujian beserta perbandingan dengan keluaran yang seharusnya. Dari sepuluh kali pengujian semua keluaran menghasilkan hasil yang sama dengan keluaran yang seharusnya.

4.5 Waktu Tempuh Setiap Alur Aplikasi

Dalam menjalankan program pada aplikasi dibutuhkan waktu untuk eksekusi program tersebut. Pengujian ini akan mengamati lama waktu eksekusi setiap alur aplikasi dan melihat efektifitas program. Pengujian ini dilakukan dengan menghitung waktu dalam milidetik pada saat alur dijalankan dan menghitung waktu pada saat alur berakhir kemudai didapatkan selisih diantara dua waktu tersebut.

4.5.1 Penentuan Tujuan

Dalam menentukan tujuan pencarian jalur, sistem menggunakan perhitungan *haversine*. Pengujian ini akan menguji berapa waktu yang dibutuhkan untuk melakukan perhitungan *haversine* hingga menggambarkan marker pada peta seperti pada tabel 4.5.

Tabel 4.5: Hasil Pengujian Waktu Eksekusi Alur Penentuan Tujuan

Pengujian ke-	Waktu Eksekusi (detik)
1	0,009
2	0,009
3	0,007
4	0,006
5	0,009
6	0,009
7	0,009
8	0,008
9	0,008
Rata-rata	0,008

Dari tabel tersebut dapat diamati bahwa waktu eksekusi alur penentuan tujuan menggunakan formula *haversine* tidak memerlukan waktu yang lama. Dari sepuluh kali pengujian didapatkan rata-rata waktu untuk eksekusi program adalah 0,008 detik.

4.5.2 Pencarian Jalur

Alur untuk mencari jalur yang digunakan untuk aplikasi ini dilakukan di *webserver*. Pengujian kali ini dilakukan di *webserver* dengan menggunakan metode yang sama seperti pengujian waktu eksekusi pada aplikasi yaitu dengan cara menghitung waktu awal fungsi dijalankan serta akhir fungsi kemudian dicari selisih waktu antar keduanya. Sehingga didapatkan hasil pengujian seperti didalam tabel 4.6.

Dari tabel hasil pengujian tersebut dapat diamati bahwa waktu eksekusi yang dibutuhkan bervariasi dan ada satu kondisi dimana waktu eksekusi lebih dari satu detik. Hal tersebut dipengaruhi oleh kecepatan dan kapabilitas server untuk mengeksekusi program tersebut. Dengan rata-rata waktu eksekusi yang hanya 0,539 detik,

Tabel 4.6: Hasil Pengujian Waktu Eksekusi Alur Pencarian Jalur

Pengujian ke-	Waktu Ekseksui (detik)
1	0,409
2	0,863
3	0,383
4	0,435
5	0,372
6	1,135
7	0,444
8	0,524
9	0,349
Rata-rata	0,539

tidak menimbulkan masalah yang serius.

4.6 Pengujian Aplikasi Lewat Kuisioner

Pengujian ini dilakukan adalah pengujian *Betha* yang dilakukan secara objektif. Pengujian dilakukan dengan kuisioner yang disusun berdasarkan faktor-faktor jaminan kualitas perangkat.

Pengujian dilakukan kepada 10 pengguna yang terbagi ke dalam 2 tipe pengguna (7 Mahasiswa dan 3 *programmer android*). Dari hasil pengujian kuisioner tersebut, dilakukan perhitungan untuk dapat menentukan kesimpulan terhadap penilaian aplikasi yang dibangun.

Kuisioner terdiri atas 10 pernyataan yang sudah disusun berdasarkan faktor-faktor yang berkaitan dengan sifat-sifat operasional *software*, yaitu faktor-faktor yang menunjukkan atribut kualitas produk dilihat dari sudut pandang pengguna, dengan lima pilihan jawaban, yaitu Sangat Tidak Setuju (STS), Tidak Setuju (TS), Kurang Setuju (KS), Setuju (S), dan Sangat Setuju (SS). Daftar pernyataan dapat dilihat pada tabel 4.7

Berdasarkan hasil pengujian *betha*, dicari presentase masing-masing jawaban dengan menggunakan rumus:

$$Y = (P/Q) \times 100\%$$

Keterangan:

Tabel 4.7: Daftar Pernyataan Dalam Kuisisioner

No.	Pernyataan
1.	Apakah anda mengetahui kondisi udara jalanan di sekitar anda terpapar polusi
2.	Kondisi udara tersebut baik untuk berolahraga
3.	Anda mengetahui daerah jalan mana yang bebas polusi
4.	Anda mengetahui ISPU (Indeks Standar Pencemaran Udara)
5.	Anda setuju apabila ada aplikasi yang dapat mencari jalan dan memandu anda melewati jalan dengan udara terbaik
6.	Anda setuju bahwa tampilan aplikasi ini mudah dimengerti
7.	Anda setuju bahwa alur aplikasi ini mudah dimengerti
8.	Visualisasi jalur dengan klasifikasi warna dapat menginformasikan anda tentang kondisi udara pada jalan
9.	Aplikasi ini dapat menginformasikan data jalan dengan kualitas udara terbaik dengan baik
10	Aplikasi ini dapat memandu anda melewati jalur dengan baik

Y = Nilai presentase

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Tabel 4.8: Presentase Responden Terhadap Pernyataan Pada Kuisisioner

No.	Pernyataan	Jumlah Responden	STS	TS	KS	S	SS
1	Pernyataan 1	10	0%	10%	30%	60%	0%
2	Pernyataan 2	10	10%	40%	40%	10%	0%
3	Pernyataan 3	10	10%	40%	20%	30%	0%
4	Pernyataan 4	10	10%	40%	20%	30%	0%
5	Pernyataan 5	10	0%	10%	0%	50%	40%
6	Pernyataan 6	10	0%	0%	30%	30%	40%
7	Pernyataan 7	10	0%	0%	10%	50%	40%
8	Pernyataan 8	10	0%	0%	20%	50%	40%
9	Pernyataan 9	10	0%	0%	30%	30%	40%
10	Pernyataan 10	10	0%	0%	0%	60%	40%

Dari tabel 4.8 didapatkan bahwa 60% responden setuju bahwa kondisi udara jalanan disekitar mereka terpapar polusi udara dan hanya 10% yang tidak setuju bahwa udara jalanan disekitarnya terpapar polusi. Kondisi udara tersebut menurut 40% responden tidak setuju dikatakan baik untuk berolahraga dan hanya 10% saja yang setuju. Sebanyak 40% dari responden tidak setuju mengetahui daerah jalan mana yang bebas polusi serta hanya 30% saja yang setuju. Mengenai Indeks Standar Pencemaran Udara (ISPU) , 40% responden tidak mengetahui informasi mengenai ISPU sedangkan 30% mengatakan mengetahui ISPU. 50% responden mendukung dibuatnya aplikasi yang dapat mencarikan jalur dan memandu pengguna melewati jalan dengan udara terbaik dan hanya 10% yang tidak setuju.

Pengujian kuisioner dari responden tentang aplikasi didapatkan hasil bahwa 40% responden setuju bahwa tampilan aplikasi ini mudah dimengerti dan hanya 10% yang mengatakan tidak setuju. Pernyataan bahwa alur aplikasi ini mudah dimengerti disetujui oleh 50% dari responden dan hanya 10 % yang tidak setuju. Mengenai visualisasi jalur dengan klasifikasi warna ini dinyatakan dapat menginformasikan tentang kondisi udara pada jalan oleh 50% responden dan hanya 20% yang kurang setuju. Aplikasi ini dinyatakan dapat menginformasikan data jalan dengan kualitas udara terbaik dengan baik oleh 40% responden dan hanya 30% yang tidak setuju. Pernyataan bahwa aplikasi ini dapat memandu pengguna dalam menyusuri jalan yang didapatkan dinyatakan setuju oleh 60% dan tidak ada responden yang tidak menyetujuinya.

BAB 5

PENUTUP

5.1 Kesimpulan

Dalam pengerjaan tugas akhir ini diterapkan beberapa fitur yang mendukung alur aplikasi pencarian dan visualisasi jalur dengan total kadar udara terbersih serta pemandu pengguna dalam melalui jalur yang didapatkan agar dapat berjalan sesuai dengan tujuan pengerjaan aplikasi.

Hasil pengujian fitur-fitur tersebut menghasilkan beberapa kesimpulan, diantaranya fitur untuk mendapatkan lokasi pengguna yang paling akurat dari ketiga layanan lokasi adalah menggunakan layanan *Fused Location Provider* yang memiliki rata-rata selisih 4,086 Meter dengan *Google Maps*. Fitur Pencarian Jalur pada sistem tugas akhir menambahkan indeks kualitas udara jalan dan jarak jalan sebagai bobot pencarian jalur, meskipun dalam pengujiannya didapatkan jarak yang lebih jauh dari pencarian jalur yang bobotnya hanya jarak saja. Akan tetapi hasil tersebut merupakan jalur dengan kualitas udara terbaik dari seluruh kemungkinan jalur menuju tujuan tersebut karena yang menjadi acuan tugas akhir ini adalah kualitas udara yang bagus bukan jarak. Aspek prioritas pada bobot jalan membantu mencari jalur dengan indeks kualitas udara yang aman bagi pengguna dalam setiap tahap. Pada Fitur penentuan arah persimpangan, metode perubahan vektor didapatkan hasil 100% akurat sesuai dengan keadaan aslinya. Pembaruan pencarian jalur dan penggambaran jalur dilakukan untuk mengantisipasi perubahan data selama navigasi berlangsung.

Dari hasil pengujian kuisioner, sebanyak 40% responden setuju bahwa tampilan aplikasi ini mudah dimengerti. Alur aplikasi ini dinyatakan mudah dimengerti oleh 50% responden. Visualisasi jalur dengan klasifikasi warna dinyatakan dapat menginformasikan tentang kondisi udara pada jalan oleh 50% responden. Aplikasi ini dinyatakan dapat memandu pengguna dalam menyusuri jalan yang didapatkan oleh 60% responden.

5.2 Saran

Pada penelitian selanjutnya akan dilakukan perubahan algoritma *dijkstra* dengan algoritma A^* . Perlu diperbaiki perumusan bobot pada tugas akhir ini, dikarenakan prioritas masih asumsi penulis yang telah diuji dengan beberapa kasus, diharapkan bobot dapat diformulasikan secara ilmiah pada pengembangan berikutnya. Selain itu, untuk penyedia layanan peta menggunakan *Open Street Map* untuk mendapatkan seluruh data koordinat persimpangan jalan. Penambahan fitur jalur alternatif dan sistem navigasi yang diperbarui setiap lokasi bergerak (*realtime*).

DAFTAR PUSTAKA

- [1] L. E. W. R. Biggs, N., “Graph theory,” Oxford University Press, 1986. (Dikutip pada halaman xiii, 6).
- [2] “How gps works.” <http://www.gps.gov/technical/ps/>. Diakses pada: 2017-03-12. (Dikutip pada halaman xiii, 9).
- [3] “Latitude and longitude.” <http://khafre.us/longitude-and-latitude>. Diakses pada: 2017-03-14. (Dikutip pada halaman xiii, 11).
- [4] Majid, “Racun karbon monoksida(co),” 2011. (Dikutip pada halaman xv, 18).
- [5] S. W. Sandri Linna Sengkey, Freddy Jansen, “Tingkat pencemaran udara co akibat lalu lintas dengan model prediksi polusi udara skala mikro,” Jurnal Ilmiah Media Engineering, vol. 1, pp. 119–126, Jul 2011. (Dikutip pada halaman 1).
- [6] W. Schwartz, “Units for use in atmospheric chemistry,” p. 67, 1995. (Dikutip pada halaman 7).
- [7] A. R. Ferdiansyah, “Penerapan algoritma dijkstra untuk menentukan rute terpendek pembacaan water meter induk pdam tirta kerta raharja kabupaten tangerang,” Jurnal TICOM, vol. 2, p. 52, Sept 2013. (Dikutip pada halaman 8).
- [8] “Gps: Theory, practice and applications.” <http://www.pdhonline.com/courses/1116/1116content.htm>. Diakses pada: 2017-03-10. (Dikutip pada halaman 9).
- [9] “Android location manager.” <https://developer.android.com/reference/android/location/LocationManager.html>. Diakses pada: 2017-04-23. (Dikutip pada halaman 12).
- [10] “Google APIs For Android fused location provider api.” <https://developers.google.com/android/reference/com/google/android/gms/location/>

- FusedLocationProviderApi. Diakses pada: 2017-03-12. (Dikutip pada halaman 13).
- [11] T. M. Korn, Grandino Arthur; Korn, Appendix B: B9. Plane and Spherical Trigonometry: Formulas Expressed in Terms of the Haversine Function. Berkeley, CA New York: Apress Distributed to the Book trade worldwide by Springer, 2000. (Dikutip pada halaman 13).
- [12] “Google maps android api.” <https://developers.google.com/maps/documentation/android-api/map?hl=id>. Diakses pada: 2017-02-17. (Dikutip pada halaman 15).
- [13] “Google maps directions api.” <https://developers.google.com/maps/documentation/directions/?hl=id>. Diakses pada: 2017-02-17. (Dikutip pada halaman 15).

BIOGRAFI PENULIS



Faishol Lukman Hakim, lahir pada 6 Nopember 1995 di Magelang, Jawa Tengah. Penulis menghabiskan masa kecilnya di Tarakan, Kalimantan Utara, lulus dari SMP Negeri 1 Tarakan pada tahun 2010 kemudian melanjutkan pendidikan menengah atasnya ke MBI Amanatul Ummah Pacet, Mojokerto hingga akhirnya lulus pada tahun 2013. Penulis kemudian melanjutkan pendidikan Strata satu ke Departemen Teknik Komputer ITS Surabaya . Saat di kuliah penulis aktif menjadi Asisten laboratorium B201 (Telematika)

hingga saat ini. Selama masa kuliah penulis sempat mengikuti beberapa perlombaan, salah satunya GO-Hackathon dan menjadi finalis. Penulis sangat tertarik dengan segala hal yang berhubungan dengan komputer dan telematika seperti aplikasi, pengolahan citra, dan berbagai ilmu telematika lainnya.

Halaman ini sengaja dikosongkan